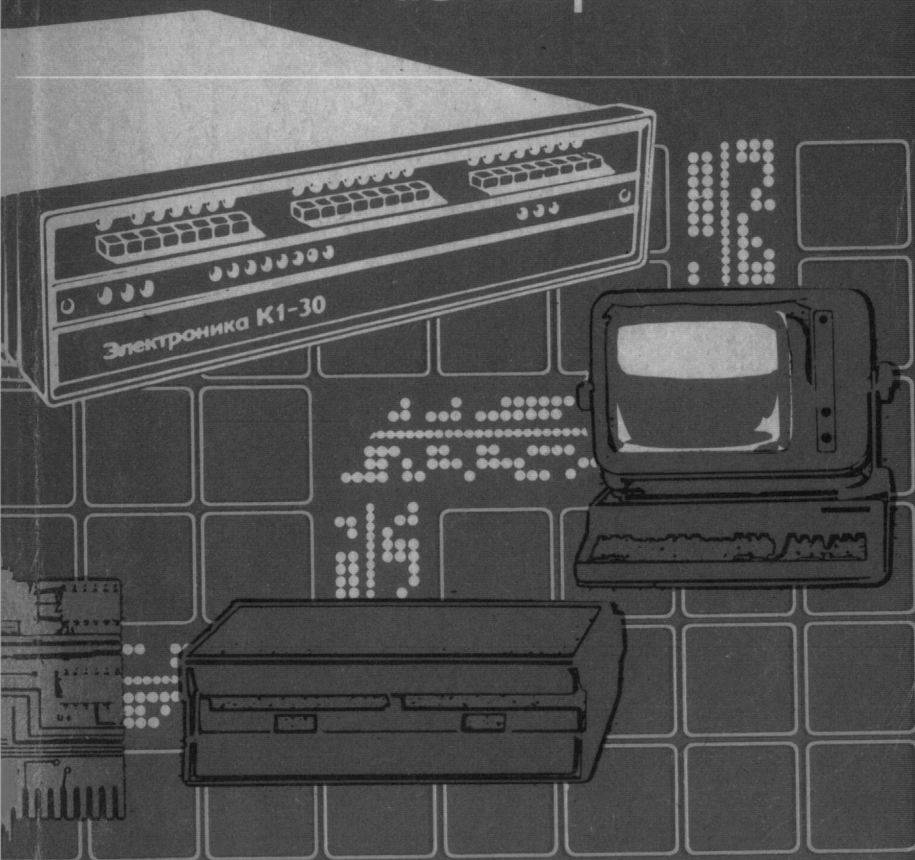


Микро- КНИГА 3

ЭВМ

Семейство ЭВМ
«Электроника К1»



Семейство ЭВМ «Электроника К1»

1

**Элементная база
микроЭВМ
«Электроника К1»**

2

**Проектирование
центральных процессоров
микроЭВМ
«Электроника К1»**

3

**Представители
микроЭВМ
«Электроника К1»**

4

**Однокристалльные
микроЭВМ
серии К1816**

5

**Программное обеспечение
комплекса технических
средств микроЭВМ
«Электроника К1-30»**

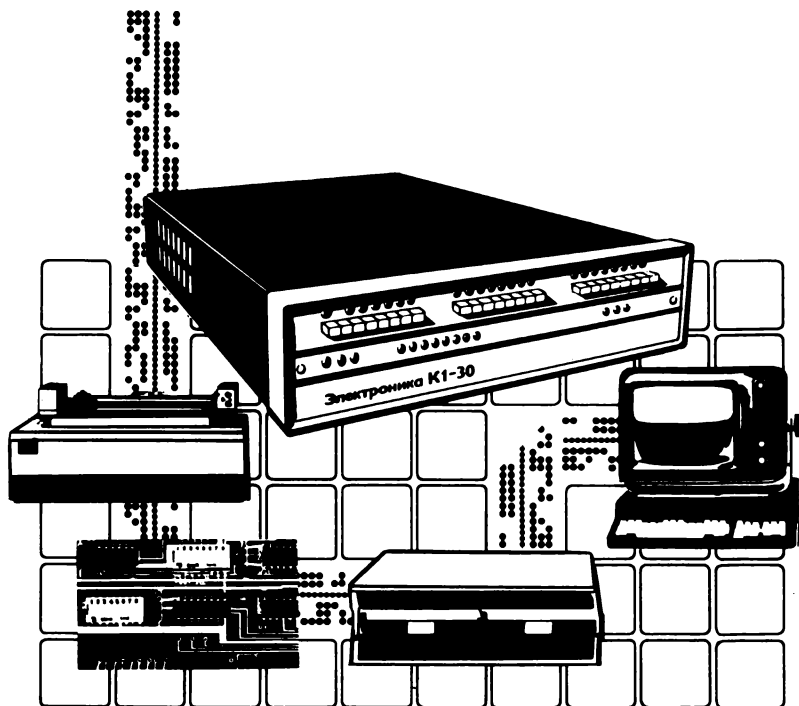
Заключение

**Перспективы развития
микроЭВМ
«Электроника К1»**

МикроЭВМ

В ВОСЬМИ КНИГАХ

Под редакцией
лауреата Государственной премии СССР
члена-корреспондента АН СССР
Л. Н. ПРЕСНУХИНА



Семейство ЭВМ «Электроника К1»

КНИГА **3**



Москва «Высшая школа» 1988

ББК 32.973.2

М 59

УДК 681.322

Рекомендовано Министерством высшего и среднего специального образования СССР для использования в учебном процессе

А. В. Кобылинский, А. В. Горячев,
Н. Г. Сабадаш, В. В. Проценко

Рецензенты: кафедра «Автоматизированные системы управления» Московского высшего технического училища им. Н. Э. Баумана (зав. кафедрой — проф. В. Н. Четвериков); лауреат Государственной премии СССР канд. техн. наук Б. И. Ермолаев (Научно-исследовательский центр электронной вычислительной техники)

МикроЭВМ: В 8 кн.: Практик. пособие/Под ред. М59 Л. Н. Преснухина. Кн. 3. Семейство ЭВМ «Электроника К1»/А. В. Кобылинский, А. В. Горячев, Н. Г. Сабадаш, В. В. Проценко. — М.: Высш. шк., 1988. — 191 с.: ил.

В пособии рассмотрены вопросы проектирования аппаратных средств и программного обеспечения микроЭВМ и контроллеров, разрабатываемых на основе микропроцессорных комплектов К580, К1810 и однокристальной ЭВМ К1816; изложены архитектурные особенности, области применения, элементная база, конструктивное исполнение, программное обеспечение и перспективы развития микроЭВМ данного семейства.

М 2405000000—190
001(01)—88 147—88

ББК 32.973.2
6Ф7.3

Издание второе, переработанное. Издательство «Высшая школа», 1988

Введение

Отечественной промышленностью выпускается широкая номенклатура средств микропроцессорной техники. Среди них по объему выпуска и степени распространения значительное место занимают микропроцессорные комплекты: 8-разрядные серии К580 и 16-разрядные серии К1810.

Микропроцессорные комплекты ориентированы на построение систем, предназначенных для решения широкого класса задач, в том числе и в реальном масштабе времени. Комплекты функционально полные, имеют в своем составе кроме центральных устройств обработки данных генераторы тактовых сигналов, магистральные формирователи, буферные регистры, контроллеры приоритетных прерываний ряда внешних устройств и прямого доступа к памяти и т. д. Микропроцессоры серий К580 и К1810 выполнены по *n*-канальной МОП-технологии, имеют достаточно высокое быстродействие и относительно небольшую потребляемую мощность.

Существенное отличие серии К1810 — возможность аппаратной реконфигурации внутренней структуры микропроцессора КМ1810ВМ86, что позволяет адаптировать его к конкретным задачам проектируемой системы. Кроме того, комплект К1810 имеет более высокое быстродействие, в два раза большую длину обрабатываемых слов, но в то же время архитектурно и программно совместим с серией К580.

Ограничения, налагаемые архитектурными и программными особенностями указанных комплектов на структуру разработанных и проектируемых микропроцессорных систем и микроЭВМ, позволяют с достаточным основанием отнести их к единому семейству — семейству «Электроника К1».

Семейство «Электроника К1» находится в непрерывном

развитии, постоянно совершенствуются его аппаратная часть и программное обеспечение. Одна из ветвей развития семейства — функционально полные однокристальные микроЭВМ серии К1816, предназначенные для применения на нижних уровнях систем управления и в разнообразных устройствах обработки данных.

Расширение номенклатуры БИС, наличие возможностей их использования для различных применений обуславливают необходимость разработки средств автоматизированной отладки программного обеспечения. Наиболее эффективные средства отладки — отладочные комплексы. Для семейства ЭВМ «Электроника К1» разработаны и успешно эксплуатируются комплексы технических средств «Электроника К1-10», «Электроника К1-30» и контроллер «Электроника К1-20».

Комплексная разработка и развитие аппаратной основы, программного обеспечения и средств отладки потенциально обуславливают широкий диапазон применения в народном хозяйстве семейства микроЭВМ «Электроника К1» как в настоящее время, так и в будущем.

глава 1

Элементная база микроЭВМ «Электроника К1»



Элементная база рассматриваемого семейства микроЭВМ отражает основную тенденцию в развитии микропроцессоров и микроЭВМ — расширение функциональных возможностей с одновременным повышением производительности на основе дальнейшего увеличения степени интеграции. В главе дается характеристика микропроцессорных комплектов К580, К1810 и, как одной из ветвей развития семейства, однокристалльных микроЭВМ серии К1816.

1.1. Общая характеристика элементной базы рассматриваемого семейства

Рост функциональной насыщенности микропроцессоров КР580ВМ80А, КМ1810ВМ86 и однокристалльной микроЭВМ КМ1816ВЕ48 характеризуется рис. 1.1. [1].

Микропроцессорные комплекты К580 и К1810 представляют собой набор функционально законченных БИС, ориентированных на модульное проектирование микроЭВМ. При этом проектирование осуществляется на основе следующих модулей (рис. 1.2): модуля центрального процессора, выполняющего основные управляющие функции в системе; модуля памяти, включающего постоянное (ПЗУ) и оперативное (ОЗУ) запоминающие устройства, предназначенные для хранения программ и данных; модуля ввода — вывода, осуществляющего сопряжение микроЭВМ с устройствами ввода — вывода.

Все модули соединяются с помощью системной магистрали, состоящей из: адресной магистрали, предназначенной для передачи адресов; магистрали данных, по которой осуществляется обмен данными; магистрали управления, предназначенной для передачи сигналов управления, которые определяют режим работы модулей.

Состав комплекта К580 иллюстрирует рис. 1.3.

Характеристики микропроцессора (МП)

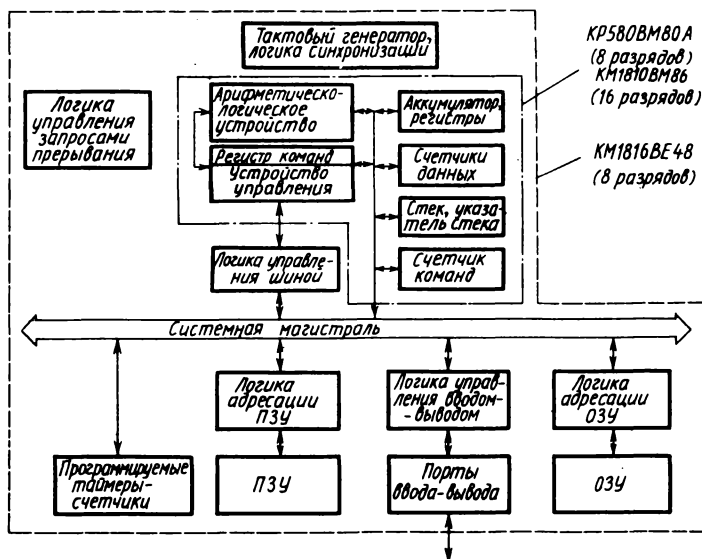


Рис. 1. Структура БИС KP580BM80A, KM1810BM86, KM1816BE

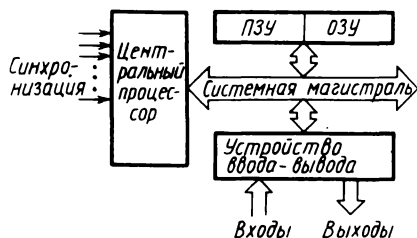


Рис. 1.2. Модульная структура микро-ЭВМ

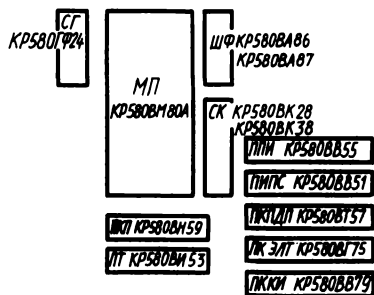


Рис. 1.3. Состав семейства KP580

KP580BM80A однозначно определяют архитектуру устройств, построенных на его основе. Фиксированная разрядность (8 двоичных разрядов), обработка слов, кратных байту, система команд, включающая широкий набор арифметических, логических команд, команд управления и пересылок, а также узел обработки прерываний и развитый интерфейс, определяют применение микропроцессора KP580BM80A как центрального процессорного устройства обработки данных. Для построения его используются

микросхемы системного генератора (СГ) КР580ГФ24 и системного контроллера магистрали (СК) КР580ВК28.

Основное назначение программируемого интерфейса последовательной связи (ПИПС) КР580ВВ51 и программируемого параллельного интерфейса (ППИ) КР580ВВ55 — коммутация и двусторонняя передача потоков информации от ЦП к устройствам ввода — вывода и наоборот. ППИ КР580ВВ55 обеспечивает коммутацию, передачу и прием параллельных потоков информации, КР580ВВ51 — прием в параллельном коде информации от ЦП, ее преобразование в последовательный поток символов и передачу на внешние устройства, а также выполняет прием от внешних устройств информации в последовательном коде, ее преобразование в параллельный и передачу к ЦП. Программируемость КР580ВВ51 и КР580ВВ55 обеспечивает возможность сопряжения практически со всеми внешними устройствами, работающими как в параллельном, так и в последовательном кодах.

Программируемый контроллер прерываний (ПКП) типа КР580ВН59 позволяет осуществлять обработку многоуровневых приоритетных прерываний в реальном масштабе времени. БИС программируемого таймера (ПТ) КР580ВН53 обеспечивает формирование программно-управляемых временных задержек и интервалов времени в микропроцессорных системах.

Программируемый контроллер прямого доступа к памяти (ПКПДП) КР580ВТ57 предназначен для организации передачи данных от быстродействующих внешних устройств в режиме прямого доступа.

Две специализированные БИС — программируемый контроллер электронно-лучевой трубки (ПК ЭЛТ) КР580ВГ75 и программируемый контроллер клавиатуры и индикации (ПККИ) КР580ВВ79 — предназначены для построения устройств вывода на экран ЭЛТ и ввода с клавиатуры и вывода на индикацию.

Основные характеристики комплекта К580 приведены ниже:

МП КР580ВМ80А

Разрядность данных	8 бит
Число команд	— 78
Разрядность команд	— 8, 16, 24

Максимальный объем адресуемой памяти	— 64 К байт
Максимальное число адресуемых внешних устройств ввода — вывода	— 256—256
Число уровней прерывания	— 8
Тактовая частота	— 2,5 МГц
Потребляемая мощность	— 1500 мВт

ППИ КР580ВВ55

Максимальное число линий для подключения внешних устройств	— 24
Число каналов передачи	— 3
Разрядность канала	— 8 бит
Потребляемая мощность	— 600 мВт

ПИПС КР580ВВ51

Разрядность последовательных данных	— 5—8 бит
Максимальная скорость обмена в режимах:	
синхронных	— 64 К байт/с
асинхронных	— 9,6 К байт/с
Тактовая частота	— 2 МГц
Потребляемая мощность	— 500 мВт

ПКП КР580ВН59

Число уровней прерывания	— 8
Возможность расширения уровней	— до 64
Потребляемая мощность	— 500 мВт

ПТ КР580ВН53

Число каналов счета	— 3
Максимальный объем счета:	
двоичного	— 2^{16}
двоично-десятичного	— 10^4
Максимальная частота счета	— 2 МГц
Потребляемая мощность	— 750 мВт

ПК КР580ВГ75

Число знаков в строке	— до 80
Число знаковых строк в кадре	— до 64
Число линий в знаковой строке	— до 16
Число градаций яркости	— 6
Максимальный период индикации линии знака	— 320
Разрядность знаков	— 8 бит
Объем ОЗУ	— $2 \times (80 \times 8)$, (16×7) бит
Потребляемая мощность	— 800 мВт

ПККИ КР580ВВ79

Число контактов клавиатуры	— 64
Разрядность приемного канала	— 8 бит
Объем СОЗУ клавиатуры	— 8×8 бит

Число индикаторов	— 2×8 или 1×16
Разрядность выходного канала	— 2×4 бит
Объем памяти индикации	— 16×8 бит
Потребляемая мощность	— 600 мВт

СГ КР580ГФ24

Максимальная частота	— 27 МГц
Потребляемая мощность	— 700 мВт

ШФ КР580ВА86 и КР580ВА87

Разрядность каналов	8 бит
Потребляемая мощность	— 800 мВт

СК КР580ВК28 и КР580ВК38

Разрядность каналов	— 8 бит
Потребляемая мощность	— 1000 мВт

ПКПДП КР580ВТ57

Число каналов ПДП	— 4
Разрядность канала	— 8 бит
Объем адресуемой памяти	— 64 К байт
Максимальная длина массива обмена	— 16 К байт
Тактовая частота	— 2 МГц
Потребляемая мощность	— 600 мВт

Кроме того, в состав комплекта входят: программируемый контроллер накопителя на гибких магнитных дисках (ПКНГМД); программируемый контроллер динамического ЗУ (ПКДЗУ); контроллер приборного интерфейса МЭК (ПИМЭК).

БИС комплекта К580 являются функционально законченными модулями, характеризующимися стандартным межмодульным интерфейсом. Некоторые вопросы построения микропроцессорных устройств, выполняемых на их основе, рассмотрены в [2].

Микропроцессорный комплект К1810 является дальнейшим развитием К580, но в отличие от него предназначен для построения микропроцессорных систем широкого диапазона применений от однопроцессорных микроконтроллеров и микроЭВМ до микропроцессорных распределенных систем.

В основу построения семейства положены три основных принципа: распределение функций микропроцессорной системы между специализированными БИС; ориентация на построение мультипроцессорных систем; иерархическая организация внутрисистемного интерфейса.

В состав комплекта (рис. 1.4) входит универсальный 16-разрядный микропроцессор КМ1810ВМ86, предназначенный для построения центральных процессоров микро-

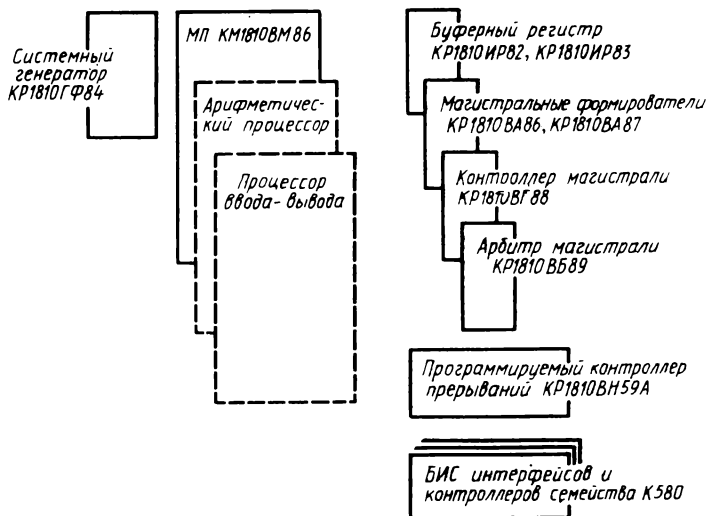


Рис. 1.4. Состав семейства К1810

ЭВМ и мультимикропроцессорных систем. Отличительной особенностью этого микропроцессора является возможность работы в двух режимах: минимальном и максимальном, — выбираемых аппаратным способом.

В минимальном режиме процессор формирует все сигналы для управления внутрисистемным интерфейсом микропроцессорной системы. Этот режим используется при построении 8- или 16-разрядных микропроцессорных контроллеров и микроЭВМ с фиксированной архитектурой. Для построения ЦП в таких устройствах требуются лишь БИС синхронизации КР1810ГФ84, буферные регистры КР1810ИР82 и КР1810ИР83 и магистральные формирователи КР1810ВА86, КР1810ВА87.

В максимальном режиме процессор формирует двоичный код, определяющий функции управления, а сигналы управления магистралью формируются специализированной БИС системного контроллера КР1810ВГ88.

Освободившиеся выходы микропроцессора используются для формирования сигналов управления, требуемых при построении мультимикропроцессорных систем.

В мультимикропроцессорных системах на базе микропроцессора КМ1810ВМ86 можно использовать процессоры двух типов: независимые и сопроцессоры.

Независимые процессоры выполняют соб-

ственный поток команд и позволяют распараллелить выполнение программы. К этому типу относятся центральные процессоры и процессоры каналов ввода — вывода.

Центральный процессор осуществляет общее управление выполнением программ. Запуск программ обычно осуществляется по сигналам прерываний. Для приоритетной обработки запросов прерываний используется программируемый контроллер прерываний КР1810ВН59А.

Процессор ввода — вывода предназначен для выполнения функций управления вводом — выводом в микропроцессорных системах.

С процессор расширяет функциональные возможности центрального процессора, т. е. форматы данных и систему команд главного процессора. К этому типу процессоров относится арифметический процессор.

Микропроцессор КМ1810ВМ86 имеет встроенные средства для координации работы процессоров обоих типов. Эти средства позволяют осуществлять разделение и отключение магистралей. Разделение реализуется средствами запроса — предоставления канала БИС КМ1810ВМ86, а также с помощью специализированной БИС — арбитра магистрали КР1810ВБ89.

Кроме того, для реализации функций интерфейса внешних устройств в микропроцессорных системах, выполненных на базе К1810, используются БИС интерфейсов и контроллеров семейства К580.

К серии К1816 относятся однокристалльные 8-разрядные микроЭВМ, представляющие собой функционально-законченные устройства, содержащие на кристалле (см. рис. 1.1) центральный процессор, ОЗУ данных и ЗУ программ, многоканальный интерфейс ввода — вывода, 8-разрядный таймер-счетчик, векторную схему прерываний, тактовый генератор, устройство синхронизации. В состав серии входят следующие БИС: КМ1816ВЕ39, КМ1816ВЕ48 и КР1816ВЕ49. Они имеют идентичную структуру и отличаются быстродействием, типом и объемом внутренней памяти программ, объемом внутреннего ОЗУ (см. гл. 5).

1.2. Микропроцессорный комплект К580

Основой построения центральных процессоров семейства «Электроника-К1», выполненных на базе серии К580, является микропроцессор КР580ВМ80А. Приведем описание функционирования БИС КР580ВМ80А с точки зрения программиста и сведения, необходимые для анализа и

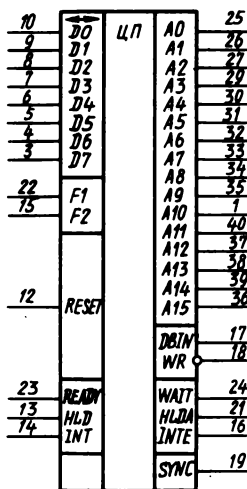


Рис. 1.5. Условное обозначение микропроцессора KP580BM80A

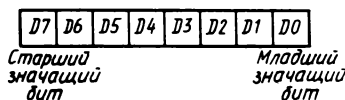
правильного понимания процесса проектирования и работы центральных процессоров семейства. Остальные данные KP580BM80A изложены в [2, 3, 5].

Микропроцессор KP580BM80A (рис. 1.5) имеет 16-разрядный канал адреса, обеспечивающий адресацию памяти объемом до 64 К байт, адресацию 256 устройств ввода и 256 устройств вывода, а также 8-разрядный канал данных.

Микропроцессор содержит внутреннее регистровое СОЗУ с организацией 6×8 или 3×16 , а также средства, позволяющие организовать в любой зоне внешней памяти стек объемом до 64 К байт. Это дает возможность обрабатывать программы практически с неограниченным числом вложений. Микропроцессор имеет средства обслуживания 8-уровне-

вых запросов прерываний, средства управления в режимах ожидания и прямого доступа к памяти, что позволяет осуществлять обмен с внешними устройствами любого быстродействия.

Структура микропроцессора ориентирована на работу с памятью, имеющую байтовую организацию. Формат обрабатываемых данных имеет вид



Байт в зависимости от типа обрабатываемой команды можно интерпретировать следующим образом:

как целое положительное число без знака в прямом коде. В этом случае числа изменяются в диапазоне 0—255:

$$0_{10} = 0000\ 0000_2 = 00_{16};$$

$$255_{10} = 1111\ 1111_2 = FF_{16};$$

как число со знаком в дополнительном коде. Старший бит числа при этом считается знаковым и кодируется 0 для положительных чисел и 1 — для отрицательных.

Остальные семь разрядов определяют величину числа. Числа изменяются в диапазоне от +127 до —128:

$$+127 = 0111\ 1111_2 = 7F_{16};$$

$$0 = 0000\ 0000_2 = 00_{16};$$

$$-1 = 1111\ 1111_2 = FF_{16};$$

$$-128 = 1000\ 0000_2 = 80_{16};$$

как *двухразрядное двоично-десятичное число без знака*. Старшие разряды D7—D4 при этом определяют старший двоично-десятичный разряд, а младшие D3—D0 — младший двоично-десятичный разряд. Диапазон представления чисел 0—99:

$$0_{10} = 0000\ 0000_2 = 00_{16};$$

$$99_{10} = 1001\ 1001_2 = 99_{16};$$

как *логический байт данных*.

Для выполнения некоторых операций допускается представление чисел в 2-байтовом формате в прямом или дополнительном коде. Числа при этом изменяются в диапазоне от +32 767 до —32 768.

Арифметические операции — сложение, вычитание, инкремент — декремент — выполняются по правилам двоичной арифметики над числами в дополнительном коде. Для обработки двоично-десятичных чисел имеется возможность преобразования результата в двоично-десятичный код. Логические операции — И, ИЛИ, ИСКЛЮЧАЮЩЕЕ ИЛИ, инверсия, сдвиги — выполняются по правилам двоичной логики.

Результат операции представляется числом в соответствующем формате и байтом признаков результата (флажков) в виде

D7	D6	D5	D4	D3	D2	D1	D0
S	Z	O	AC	O	P	1	C

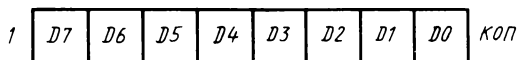
Здесь 5 бит признаков устанавливаются в зависимости от результатов выполнения операции следующим образом: бит знака S устанавливается равным биту 7 (D7) результата; бит нуля Z устанавливается в 1, если результат равен 0, в противном случае сбрасывается в 0; бит четности P устанавливается в 1, если число единиц в результате четно, в противном случае сбрасывается в 0; бит переноса C устанавливается в 1, если при выполнении операции сложения возникает перенос из старшего разряда или при выполнении вычитания

перенос не возникает (т. е. происходит заем из старшего разряда), в противном случае устанавливается в 0; бит дополнительного переноса АС устанавливается в 1 при появлении переноса из бита 3 (D3) в бит 4 (D4) результата при выполнении операций сложения и вычитания. Признак используется для преобразования двоичного результата в двоично-десятичный при обработке двоично-десятичных чисел.

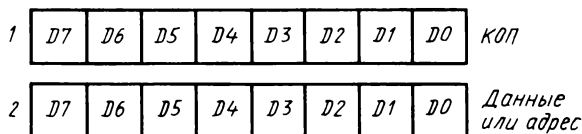
В свою очередь, формат представления команд зависит от типа команды и используемого способа адресации. Для представления кода операции используется 1 байт. Процессор имеет возможность прямой адресации до 65 536 байт памяти, для чего используется адрес длиной 16 бит. Максимальная длина команды составляет 3 байта.

Для обеспечения гибкости кодирования вычислительного процесса используются три формата команд:

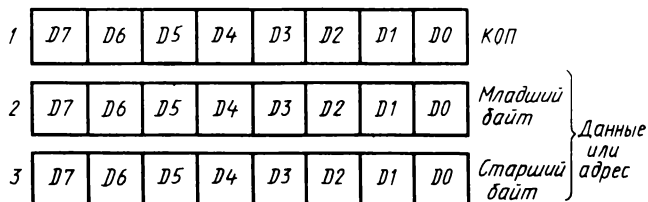
байт. 1- байтовый формат



2- байтовый формат



3- байтовый формат



Многобайтовые команды хранятся в соседних ячейках памяти и адресуются по байту 1. Формат определяется кодом операции (КОП).

Способы адресации, используемые в микропроцессоре КР580ВМ80А, следующие:

непосредственная — байт 2 или 2 и 3 непосредствен-

но содержат данные (операнд или адрес, заносимый в регистр). Единственное исключение составляет команда вызова подпрограммы обслуживания прерывания — адрес вектора указывается в байте КОП;

прямая — для адресации слов данных, содержащихся в памяти или во внешнем устройстве. Прямой адрес указывается в байтах 2 или 2 и 3 команды;

прямая регистровая — для адресации 1- и 2-байтовых слов, содержащихся в регистрах процессора (адреса регистров, участвующих в операции, указываются в байте КОП);

косвенная регистровая — для адресации байтов данных, находящихся в памяти, адрес ячеек которой определяется содержимым регистров, используемых в качестве адресных указателей (адрес регистра указывается в байтах КОП);

стековая — для косвенной адресации 2-байтовых слов данных или адресов, находящихся в области памяти,

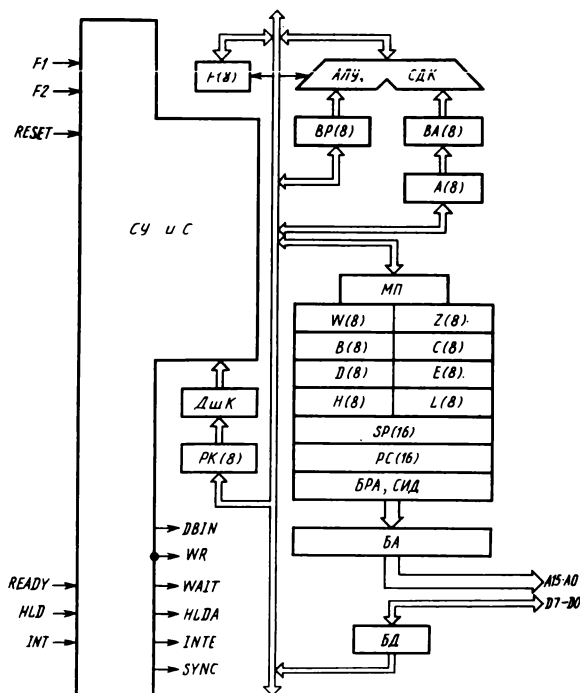


Рис. 1.6. Функциональная схема МП КР580ВМ80А

отведенной под стек (адрес определяется по содержимому регистра — указателя стека);

неявная — адрес регистра в команде явно не указывается, а способ адресации задается КОП.

Микропроцессор КР580ВМ80А реализован на основе общей внутренней магистрали данных и включает в себя следующие функциональные узлы (рис. 1.6): блок регистров с адресной логикой; блок АЛУ; двунаправленную буферизованную магистраль данных; блок управления и синхронизации (1, 4).

Блок регистров содержит шесть 16-битовых регистров, образующих статическую память с произвольным доступом. Три из них могут использоваться как шесть отдельных 8-разрядных программно-доступных регистров В, С, D, Е, H, L общего назначения для хранения операндов или как три 16-разрядные программно-доступные пары ВС, DE, HL для хранения адресов или 2-байтовых операндов. При выполнении арифметических и логических бинарных операций с регистровой адресацией в регистрах В, С, D, Е, H, L хранятся 8-разрядные операнды, которые передаются в блок АЛУ для участия в операции. Второй операнд и результат операции сохраняются в блоке АЛУ. При выполнении операций инкремент — декремент регистры используются в качестве источников данных и приемников результата операции.

Пары регистров ВС, DE, HL обычно используются в качестве регистров-указателей косвенной адресации при выполнении арифметических, логических и пересылочных операций. При этом основным регистром-указателем является пара HL. Исключение составляет операция сложения 2-байтовых слов, при выполнении которой пары ВС, DE, HL используются для хранения 16-разрядных операндов. В этом случае содержимое любой пары суммируется в блоке АЛУ с содержимым пары HL и результат записывается в HL.

Содержимое каждого из регистров можно переслать в блок АЛУ или в память через 8-битовые мультиплексоры (МП) и внутреннюю магистраль данных. Аналогично происходит загрузка регистров.

Содержимое каждой пары регистров программно можно увеличить или уменьшить на 1. Изменение осуществляется с помощью адресной логики.

Регистр-счетчик РС используется в качестве программного счетчика и хранит адрес текущей команды программы. Его содержимое автоматически увеличивается

после выборки каждого байта команды схемой адресной логики. Загрузка и выдача содержимого РС осуществляются через мультиплексоры и внутреннюю магистраль данных.

Указатель стека SP хранит адрес ячейки области памяти, к которой было сделано последнее обращение. Содержимое указателя стека уменьшается на 1 перед каждым занесением слова в стек или увеличивается на 1 после каждого извлечения слова из стека с помощью адресной логики. Указатель загружается с внутренней магистрали данных через мультиплексор. Содержимое можно прибавить к содержимому HL, для чего имеется возможность пересылки содержимого SP через мультиплексоры на внутреннюю магистраль данных.

Пара регистров WZ используется для временного хранения байтов 2 и 3 команд перехода, передаваемых с внутренней магистрали данных в программный счетчик.

Адресная логика предназначена для хранения программного изменения и выдачи на магистраль A15—A0 адресов данных и команд. Она содержит буферный регистр адреса (БРА), логическую схему инкремента — декремента (СИД) и адресный буфер (БА). Буферный регистр адреса принимает и хранит адрес с любого 16-битового регистра. Его выход связан со входами СИД и БА.

Схема инкремента-декремента представляет собой схему быстрого переноса (заема), с ее помощью содержимое БРА можно передать с изменением на +1, —1 или без изменения через 16-разрядный мультиплексор на выход любого 16-битового регистра BC, DE, HL, SP или PC.

Адресный буфер представляет собой 16 выходных формирователей с тремя состояниями и предназначен для выдачи адреса на выводы адресной магистрали A15—A0. Наличие третьего (высокоимпедансного) состояния в регистре адреса позволяет непосредственно подключить микропроцессор к общей системной адресной магистрали микроЭВМ.

Б л о к АЛУ предназначен для выполнения арифметических и логических операций над числами в параллельном 8-разрядном двоичном коде. Информация обрабатывается в АЛУ с использованием регистра временного хранения (ВР), аккумулятора временного хранения (ВА), аккумулятора (А) и регистра признаков (F). При выполнении бинарных операций один из операндов пересылается из аккумулятора в регистр ВА; второй — посту-

пает из памяти или блока регистров через внутреннюю магистраль данных в регистр ВР, а затем передается в АЛУ через кодопреобразователь в прямом или обратном коде в зависимости от выполняемой операции. Результат операции передается через внутреннюю магистраль в аккумулятор или регистр общего назначения, а признаки результата записываются в регистр признаков F. Обмен содержимого аккумулятора и регистра признаков с памятью или блоком регистров осуществляется также через внутреннюю магистраль данных.

Схема десятичной коррекции (СДК) предназначена для преобразования двоично-кодированного результата в двоично-десятичный код при обработке двоично-десятичных чисел.

Двунаправленная магистраль данных служит для организации связи микропроцессора с другими микросхемами, входящими в состав микроЭВМ. Она включает в себя внутреннюю магистраль данных, буфер данных (БД) и соединена с выводами магистрали данных D7—D0 процессора. Буфер данных—8-разрядный двунаправленный с тремя состояниями—предназначен для развязки внутренней и внешней магистралей данных. Он состоит из буферного регистра данных и формирователей. В режиме вывода информация с внутренней магистрали загружается в буферный регистр, а затем передается на внешнюю магистраль данных через формирователи. При вводе данные из внешней магистрали через формирователи непосредственно передаются на внутреннюю магистраль. Буферный регистр данных при этом отключается. Он отключается также при выполнении операций, не связанных с передачей информации процессора.

Блок управления и синхронизации предназначен для приема команд, синхронизирующих и управляющих внешних сигналов, а также для формирования внутренних сигналов микроопераций и внешних синхронизирующих и управляющих сигналов. Он содержит регистр команд (РК), дешифратор команд (ДшК), схемы формирования машинных циклов и другие устройства. РК хранит байт кода операции, поступающей с внутренней магистрали данных. Его содержимое после дешифрации и синхронизации времязадающими сигналами используется для формирования сигналов внутреннего управления блоком регистров, АЛУ с регистрами и буфером данных, реализующих микропрограммы выполнения команд. Прием и выдача синхронизирующих

и управляющих сигналов осуществляются через выводы микропроцессора (рис. 1.7.).

Выводы синхронизации:

F1, F2 — входы двух непрерывно сменяющихся последовательностей синхронимпульсов;

SYNC — выход сигнала синхронизации, определяющего начало каждого машинного цикла команды. Используется для идентификации действий, выполненных в микропроцессоре, и синхронизации поциклового управления процессором;

RESET — вход сигнала начальной установки процессора. После окончания сигнала (его длительность должна быть не менее трех периодов синхросигналов) программный счетчик устанавливается в нулевое состояние и процессор начинает выполнять программу с нулевого адреса.

Выводы управления ожиданием:

READY — вход сигнала готовности, указывающего готово (READY=1) или нет (READY=0) внешнее устройство или память к обмену. Позволяет синхронизировать обмен информацией с внешними устройствами или с памятью более низкого быстродействия. Может использоваться для организации пошагового, покомандного или с остановом по требуемому адресу режимов работы микропроцессора;

WAIT — выход сигнала, указывающего на то, что процессор находится в состоянии ОЖИДАНИЕ (WAIT=1).

Выводы управления памятью:

WR — выход сигнала управления записью в память или во внешнее устройство; низкий активный уровень сигнала указывает на то, что микропроцессор выдал данные на магистраль D7—D0.

Выводы управления магистралью данных:

DBIN — выход сигнала, указывающего на то, что магистраль данных (D7—D0) находится в режиме приема. Используется для управления чтением данных из памяти или из внешних устройств.

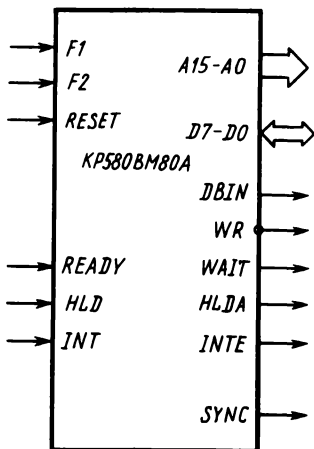


Рис. 1.7. Схема подключения выводов МП КР580ВМ80А

Выводы управления прерываниями:

INT — вход сигнала запроса прерывания, поступающего от внешних устройств, запрашивающих обмен с микропроцессором в режиме прерывания;

INTE — выход сигнала разрешения прерывания ($INTE=1$), указывающего на то, что процессор готов к обмену в режиме прерывания (может принять запрос прерывания). Используется для управления цепями формирования запроса прерывания. Если процессор готов к обмену в режиме прерывания, то на выходе устанавливается сигнал высокого уровня и процессор может принять запрос прерывания. После перехода к обслуживанию прерывания на выходе **INTE** устанавливается сигнал низкого уровня и запросы прерывания не воспринимаются. Способность восприятия процессором последующих запросов устанавливается программным путем.

Выводы управления захватом магистралей в режиме ПДП:

HLD — вход сигнала запроса захвата магистралей **D7—D0** и **A15—A0** внешними устройствами. Предназначается для управления в режиме прямого доступа к памяти. Под действием сигнала захвата процессор переходит в состояние **ЗАХВАТ**, в котором магистрали данных и адреса микропроцессора находятся в третьем состоянии, благодаря чему системная магистраль может использоваться внешними устройствами;

HLDA — выход сигнала подтверждения захвата, указывающего на то, что процессор находится в состоянии **ЗАХВАТ**. Магистрали данных и адреса процессора при этом отключены от выходов.

С точки зрения пользователя и разработчика центрального процессора микроЭВМ важными являются сведения о системе команд микропроцессора **KP580BM80A**. В этом смысле микропроцессор представляет собой массив программно-доступных регистров, над содержанием которых выполняются указанные в командах операции.

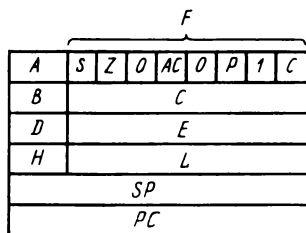
Программно-доступными являются:

8-битовые регистры **A, B, C, D, E, H, L**;

16-битовые — слово состояния **PSW** (регистры **A** и **F**), пары регистров **BC, DE, HL**, указатель стека **SP**, программный счетчик **PC**;

отдельные признаки регистра **F**;

триггер разрешения прерывания **INTE**;



☐ INTE

В зависимости от характера выполняемых операций систему команд микропроцессора КР580ВМ80А можно разбить на следующие группы: команды пересылки; арифметические и логические команды; команды управления.

Группа команд пересылки содержит следующие команды:

MOV D, S — пересылки между регистрами А, В, С, D, Е, H, L или между регистрами и памятью с использованием соответственно регистровой и косвенной адресации через пару HL;

MVI D, D8 — непосредственной загрузки регистров или ячеек памяти байтом данных, содержащимся в байте 2 команды;

LDA ADR — загрузки аккумулятора из памяти по адресу, указанному в адресной части команды;

STA ADR — запоминания содержимого аккумулятора в памяти по адресу, заданному в адресной части;

LDAX rp — загрузки аккумулятора из памяти по адресу, указанному в паре регистров ВС или DE;

STAX rp — запоминания содержимого аккумулятора в памяти по адресу, указанному в паре регистров ВС или DE:

MOV D, S

01	DDD	SSS
----	-----	-----

DDD, SSS = 000 - В
001 - С
010 - D
011 - E
100 - H
101 - L
110 - M(HL)
111 - A

STAXrp
LDAXrp

00	rp	0/1	0/0
----	----	-----	-----

rp = 00 - В
01 - D

MVI D, D8

00	DDD	110
----	-----	-----

D8

STAADR
LDAAADR

0	0	1	1	0/1	0	1	0
---	---	---	---	-----	---	---	---

ADR (7-0)

ADR (15-8)

LHLD ADR — загрузки регистровой пары HL двумя байтами из памяти, адрес первого из которых задан в байтах 2 и 3 команды, а адрес второго вычисляется увеличением на 1 адреса первого;

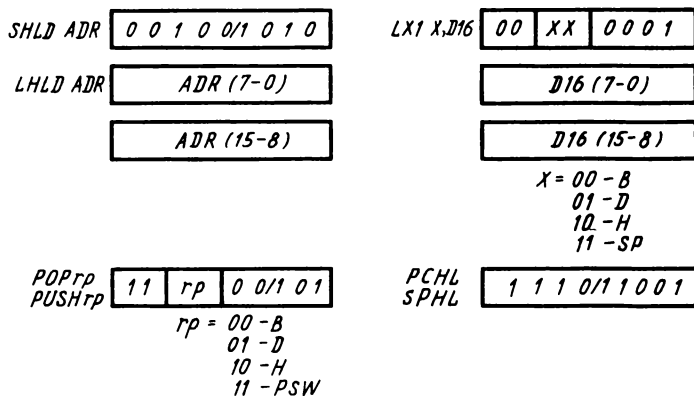
SHLD ADR — запоминания содержимого регистровой пары HL в двух байтах памяти, адрес младшего из которых задан в байтах 2 и 3 команды, а адрес старшего вычисляется увеличением на 1 адреса младшего байта;

LXI X, D16 — непосредственной загрузки регистровых пар и указателя стека 2-байтовыми словами, содержащимися в байтах 2 и 3 команды;

PUSH гр — занесения в стек 2 байт данных из регистров BC, DE, HL или PSW;

POP гр — извлечения из стека 2 байт данных и занесения их в регистры BC, DE, HL или PSW;

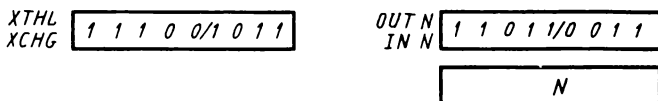
SPHL и **PCHL** — пересылки 2-байтовых слов из регистровой пары HL в указатель стека или программный счетчик. Первая команда позволяет изменить адрес области памяти, отведенной под стек, а вторая — осуществлять относительные переходы:



XCHG и **XTHL** — обмена 2-байтовыми словами между регистровыми парами HL и DE или содержимым верхушки стека и пары HL;

IN N — ввода байта данных в аккумулятор;

OUT N — вывода байта данных из аккумулятора;



Группа арифметических и логических команд содержит следующие команды:

ADD S и **ADI D8** — сложения содержимого аккумулятора с содержимым РОН или ячейки памяти, адресуемой парой HL, или непосредственно с байтом данных, содержащимся в байте 2 команды; при этом результат сохраняется в аккумуляторе и устанавливаются признаки;

ADC S и **ACI D8** — сложения содержимого аккумулятора с прибавлением бита переноса с парами регистров или ячейкой памяти, указанной в паре HL, или непосредственно с байтом данных, содержащимся в байте 2 команды; позволяют реализовать сложение многобайтовых слов;

SUB S и **SUI D8** — вычитания из содержимого аккумулятора байта данных из регистра общего назначения или ячейки памяти, косвенно адресуемой содержимым пары HL или непосредственно байта 2 команды;

SBB S и **SBI D8** — вычитания с заемом с использованием содержимого триггера переноса, что позволяет выполнять вычитание многобайтовых слов;

ANA S, **ORA S** и **XRA S** — реализующие логические операции И, ИЛИ, ИСКЛЮЧАЮЩЕЕ ИЛИ над содержимым аккумулятора и одним из РОН или ячейки памяти, косвенно адресуемой содержимым пары HL;

ANI D8, **ORI D8**, **XRI D8** — реализующие логические операции И, ИЛИ, ИСКЛЮЧАЮЩЕЕ ИЛИ над содержимым аккумулятора и байтом 2 команды; как и в предыдущей команде, результат остается в аккумуляторе;

CMP S — сравнения содержимого аккумулятора и одного из РОН или ячейки памяти, адресуемой парой регистров;

CPI D8 — сравнения содержимого аккумулятора с байтом 2 команды; содержимое аккумулятора, как и в предыдущем случае, не изменяется, а устанавливаются флажки:

1 0	КОП	SSS
ADD S	000	000 - B
ADC S	001	001 - C
SUB S	010	010 - D
SBB S	011	011 - E
ANA S	100	100 - H
XRA S	101	101 - L
ORA S	110	110 - M(HL)
CMP S	111	111 - A

0 0	КОП	1 1 0
D8		
ADI	D8	000
ACI	D8	001
SUI	D8	010
SBI	D8	011
ANI	D8	100
XRI	D8	101
ORI	D8	110
CPI	D8	111

INR D и INX X — увеличения содержимого 1-байтовых регистров A, B, C, D, E, H, L или ячеек памяти, адресуемых парой HL, и 2-байтовых регистров BC, DE, HL и SP на 1;

DCR D и DCX X — уменьшения содержимого 1-байтовых регистров A, B, C, D, E, H, L и 2-байтовых регистров BC, DE, HL и SP на 1;

DAD X — сложения содержимого одной из пар регистров BC, DE, SP с содержимым пары HL; результат при этом сохраняется в HL, устанавливается признак переноса;

DAA — десятичной коррекции, с помощью которой результат двоичного сложения двоично-десятичных чисел преобразуется в двоично-десятичный код:

INR D 0 0 D D D 1 0 0/1
DCR D

D = 000 - B
001 - C
010 - D
011 - E
100 - H
101 - L
110 - M
111 - A

INX X 0 0 X X 0/1 0 1 1
DCX X

X = 00 - B
01 - D
10 - H
11 - SP

DAA 0 0 1 0 0 1 1 1

DAD X 0 0 X X 1 0 0 1

X = 00 - B
01 - D
10
11

RLC, RRC или RAR, RAL — циклического сдвига содержимого аккумулятора на один разряд влево или вправо с заполнением флага переноса или с включением флага переноса в цепь сдвига;

CMA, CMC, STC — инверсии каждого бита аккумулятора (CMA) или флага переноса (CMC) и установки в 1 флага переноса (STC):

0 0 0 КОП 1 1 1

RLC 00
RRC 01
RAL 10
RAR 11

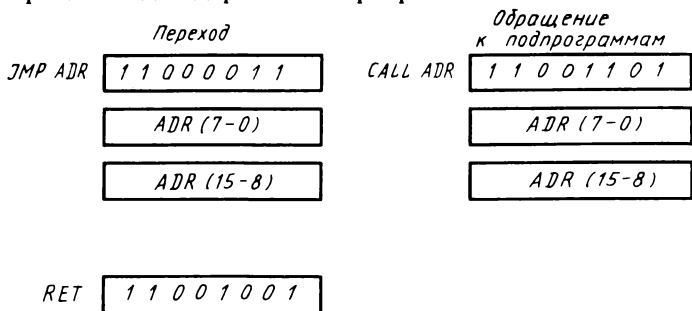
0 0 1 КОП 1 1 1

CMA 01
STC 10
CMC 11

Группа команд управления содержит команды:

JMP ADR, CALL ADR и RET — безусловного перехода по адресу; вызов подпрограммы и возврат из

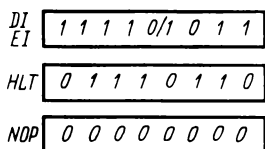
подпрограммы, при выполнении которых осуществляются операции над содержимым программного счетчика:



ЕІ или DІ — установки или сброса триггера разрешения прерывания, осуществляющие программное разрешение или запрет прерываний;

HLT — остановка выполнения команд в процессоре;

NOP — холостую команду, не вызывающую изменения состояния регистров процессора, кроме добавления 1 в программный счетчик:



Команды условного перехода вызывают передачу управления при выполнении условий переходов, определяемых значениями 4 бит условий (Z, C, P, S). Условия перехода и их мнемонические обозначения приведены в табл. 1.1.

Таблица 1.1

Условие перехода	Мнемоническое обозначение	Код CCC
Неравенство нулю (Z = 0)	NZ	000
Равенство нулю (Z = 1)	Z	001
Отсутствие переноса (C = 0)	NC	010
Наличие переноса (C = 1)	C	011
Нечетность (P = 0)	PO	100
Четность (P = 1)	PE	101
Положительность (S = 0)	P	110
Отрицательность (S = 1)	M	111

Форматы команд условного перехода имеют вид

<div>11</div>	<div>CCC</div>	<div>010</div>	<div>11</div>	<div>CCC</div>	<div>100</div>
<div>ADR (7-0)</div>			<div>ADR (7-0)</div>		
<div>ADR (15-8)</div>			<div>ADR (15-8)</div>		
<i>JNZ</i>	<i>ADR</i>	<i>C= 000</i>	<i>CNZ</i>	<i>C= 000</i>	
<i>JZ</i>	<i>ADR</i>	<i>001</i>	<i>CZ</i>	<i>001</i>	
<i>JNC</i>	<i>ADR</i>	<i>010</i>	<i>CNC</i>	<i>010</i>	
<i>JC</i>	<i>ADR</i>	<i>011</i>	<i>CC</i>	<i>011</i>	
<i>JPO</i>	<i>ADR</i>	<i>100</i>	<i>CPO</i>	<i>100</i>	
<i>JPE</i>	<i>ADR</i>	<i>101</i>	<i>CPE</i>	<i>101</i>	
<i>JP</i>	<i>ADR</i>	<i>110</i>	<i>CP</i>	<i>110</i>	
<i>JM</i>	<i>ADR</i>	<i>111</i>	<i>CM</i>	<i>111</i>	

<div>11000000</div>
<i>RNZ</i> <i>C= 000</i>
<i>RZ</i> <i>001</i>
<i>RNC</i> <i>010</i>
<i>RC</i> <i>011</i>
<i>RPO</i> <i>100</i>
<i>RPE</i> <i>101</i>
<i>RP</i> <i>110</i>
<i>RM</i> <i>111</i>

В командах переходов в программный счетчик заносится адрес, содержащийся в байтах 2 и 3 команды. В командах возвращения к подпрограмме вначале адрес следующей команды (адрес возврата), содержащийся в программном счетчике, сохраняется в стеке, а затем заменяется в РС адресом начала подпрограммы из байтов 2 и 3 команды. В командах безусловной передачи управления эти операции выполняются всегда, а в командах условной — при выполнении условия перехода;

RST V — перехода на подпрограмму обслуживания прерываний; обеспечивает обращение к восьми подпрограммам обслуживания прерывания:

<i>RST V</i>	<div>11</div>	<div>VVV</div>	<div>111</div>
	<i>000</i>	<i>- RST0</i>	
	<i>001</i>	<i>- RST1</i>	
	<i>010</i>	<i>- RST2</i>	
	<i>011</i>	<i>- RST3</i>	
	<i>100</i>	<i>- RST4</i>	
	<i>101</i>	<i>- RST5</i>	
	<i>110</i>	<i>- RST6</i>	
	<i>111</i>	<i>- RST7</i>	

адрес указывается кодом VVV (вектором прерывания) в разрядах 4, 5, 6 КОП. Адреса перехода зависят от кода VVV:

Код	Адрес подпрограммы	Код	Адрес подпрограммы
000 ₂	0000 ₁₀	100 ₂	0032 ₁₆
001 ₂	0008 ₁₀	101 ₂	0040 ₁₀
010 ₂	0016 ₁₀	110 ₂	0048 ₁₀
011 ₂	0024 ₁₆	111 ₂	0056 ₁₀

Каждая из приведенных команд в микропроцессоре выполняется в течение командного цикла, включающего в себя циклы выборки и исполнения команды. Команды микропроцессора КР580ВМ80А состоят из различного числа байтов, а соответственно и длительность цикла выборки для разных команд различна. Длительность цикла выполнения команды также зависит от типа команды и способа адресации операндов в команде. В основном длительность командного цикла в микропроцессоре КР580ВМ80А определяется числом обращений к памяти или внешнему устройству.

Интервал времени, в течение которого осуществляется одно обращение процессора к памяти или внешнему устройству, определяется как машинный цикл М. Командный цикл процессора состоит, таким образом, из того или иного (в зависимости от типа команды) числа машинных циклов.

Число машинных циклов для команд процессора изменяется от 1 для 1-байтовых команд с регистровой адресацией до 5 для 3-байтовых сложных команд.

Машинный цикл, в свою очередь, разбивается на некоторое количество машинных тактов T , в течение каждого из которых выполняется элементарное действие (микрооперация) в процессоре. Число тактов в цикле определяется кодом команды и изменяется от 3 до 5. Длительность такта зависит от частоты импульсной последовательности, формируемой внешними цепями.

Для синхронизации процессора с памятью и внешними устройствами с меньшим быстродействием при организации работы в режиме прямого доступа к памяти и останова процессора предусмотрены три специальных состояния: ОЖИДАНИЕ, ЗАХВАТ и ОСТАНОВ, — длительности которых произвольны, но всегда кратны длительности такта T .

Таким образом, командный цикл микропроцессора КР580ВМ80А состоит из некоторого числа машинных

циклов, а каждый машинный цикл — из определенного числа тактов, в течение которых выполняются те или иные микрооперации в процессоре.

Типы машинных циклов, временные диаграммы выполнения команд, а также режимы работы микропроцессора достаточно подробно изложены в [2,3].

1.3. Микропроцессорный комплект КМ1810

Рассмотрим основные технические характеристики микропроцессора КМ1810ВМ86, являющегося основой комплекта КМ1810.

Микросхема КМ1810ВМ86 представляет собой 16-разрядный микропроцессор, выполненный по *n*-канальной МОП-технологии. Особенности его являются: наличие свойств 8- и 16-разрядных процессоров; возможность обработки слов в двоичном, двоично-десятичном кодах, строк данных и отдельных битов; наличие встроенных средств выполнения операций умножения и деления чисел [6].

Общий объем непосредственно адресуемой памяти — 1 М байт. При обращении к памяти происходит ее сегментация, что обеспечивает модульное проектирование программного обеспечения и упрощает программирование и отладку.

Процессор имеет: 16-разрядный канал данных; внутреннее СОЗУ, состоящее из четырнадцати 16-разрядных регистров с симметричными операциями. Микропроцессор обладает возможностью прямой адресации до 64 К устройств ввода и до 64 К устройств вывода, имеет гибкую структуру средств обработки внутренних, программных и внешних прерываний. Максимальное число типов прерываний составляет 256.

Высокая производительность процессора обеспечивается наличием работающих с одинаковой частотой и выполняющих операции одновременно отдельных устройств обработки команды и средств ее выборки.

Микропроцессор синхронизируется тактовыми импульсами частотой 5 МГц, имеет один источник питания +5В.

Структурная организация КМ1810ВМ86 обеспечивает возможность использования средств пользователя и систем разработанных на базе микропроцессора КР580ВМ 80А. Большинство существующих программ для микропроцессорных средств вычислительной техники, реализованных на базе КР580ВМ80А, можно преобразовать для систем, выполненных на основе КМ1810ВМ86.

Структура микропроцессора имеет вид, показанный на рис. 1. 8. Существенным является наличие устройств: обработки информации (УО), сопряжения с каналом (УСК), управления и синхронизации (УУС).

Устройство обработки информации предназначено для обработки данных и формирования исполнительных адресов. В состав его входят: 16-разрядное АЛУ, микропрограммное устройство управления и регистры общего назначения. АЛУ выполняет арифметические и логические операции с 8- и 16- разрядными операндами, результат обработки фиксируется в одном из регистров или передается в УСК по двунаправленной

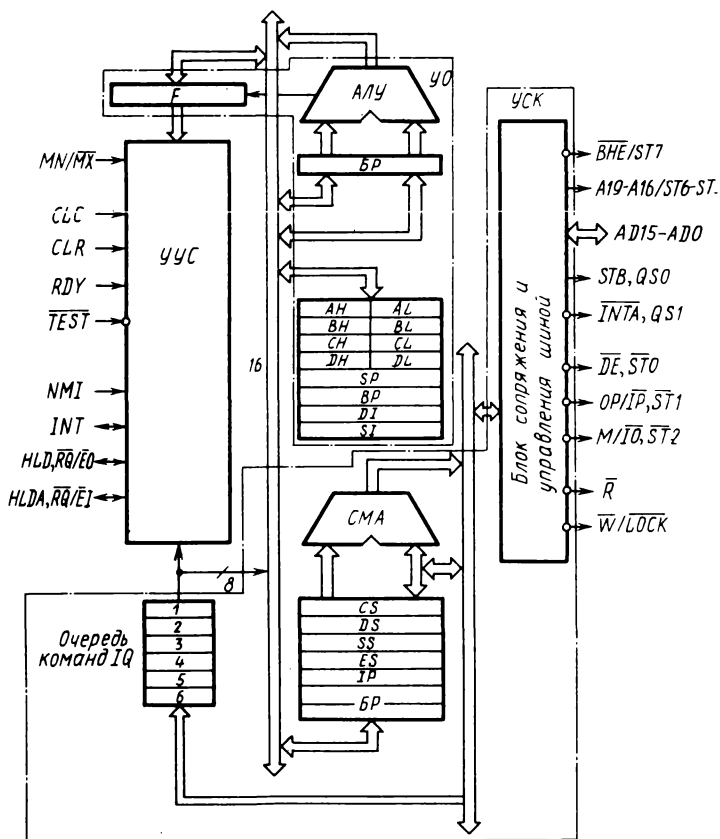
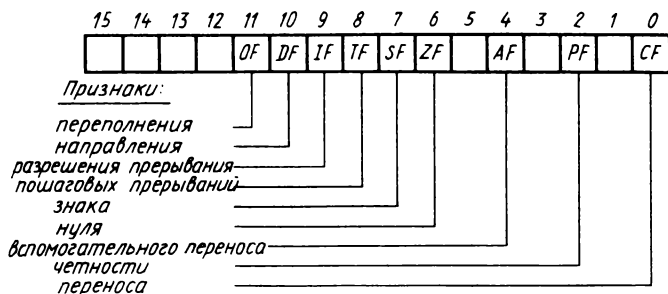


Рис. 1.8. Структура МП KM1810BM86

16-разрядной магистрали. По результатам формируются признаки, которые записываются в регистр F:



Микропрограммное устройство управления обработки представляет собой совокупность микропрограммного устройства управления и аппаратно реализованных средств управления выполнением отдельных, наиболее критичных по времени операций.

В микропроцессоре имеются программно доступные регистры трех типов (рис. 1.8):

- регистры общего назначения AH, AL, BH, BL, CH, CL, DH, DL, SP, BP, DI, SI;
- регистр признаков F;
- сегментные регистры CS, DS, SS, ES.

Регистр адреса команд IP 16-разрядный программно-недоступный и используется в командах передачи управления.

Регистры AH, AL, BH, BL, CH, CL, DH, DL образуют группу HL регистров данных и используются при выполнении арифметических и логических команд. Указанные регистры можно адресовать как четыре 16-разрядных или восемь 8-разрядных регистров.

Регистры SP, BP, DI, SI адресуются только как 16-разрядные, являются регистрами-указателями и индексными. Используются для хранения адреса смещения внутри текущего сегмента стека или данных при вычислении адреса, а также при выполнении арифметических и логических команд.

Сегментные регистры CS, DS, SS и ES предназначены для хранения базовых адресов (адреса начальных ячеек сегментов) и позволяют адресовать память объемом до 1 М байт путем деления ее на сегменты объемом до 64 К байт.

Содержимое регистра CS определяет текущий сегмент

кода. Все команды выбираются из памяти относительно данного сегмента, а содержимое регистра IP используется как смещение.

Содержимое регистра DS определяет текущий сегмент данных. Однако при использовании перед командой 1-байтового префикса замены можно осуществлять адресацию данных относительно одного из трех остальных сегментов (CS, ES, SS).

Содержимое регистра SS определяет текущий сегмент стека. При использовании 1-байтового префикса указателя базы (регистр BP) адресация осуществляется относительно одного из трех оставшихся сегментов.

Устройства обработки информации не имеют непосредственной связи с внешней памятью. Обмен с памятью и устройствами ввода — вывода возможен через УСК.

Устройство сопряжения с каналом выполняет операции обмена с внешней памятью, выборки команд и загрузки их в очередь, а также операции циклов обмена, запрашиваемых УО. Запрос УО обмена имеет приоритет перед запросом цикла выборки команды. При выполнении команд перехода, вызова или прерывания УСК прекращает выборку команд, устанавливает в исходное состояние счетчик загрузки и выборки очереди команд, восстанавливает содержимое регистра адреса команды и заполняет очередь команд по новому адресу. Таким образом, при последовательном выполнении команд программ очередь заполняется непрерывно. При передаче управления УСК сбрасывает очередь, выбирает команду по адресу перехода, передает ее на исполнение, а затем вновь заполняет очередь.

Для хранения пересылаемых данных и адресов в составе УСК имеется два буферных регистра (БР). Обмен с внешней памятью и устройствами ввода — вывода осуществляется через выходные магистрали (рис. 1.9):

$\overline{BHE}/ST7$ — выход разрешения передачи старшего байта дан-

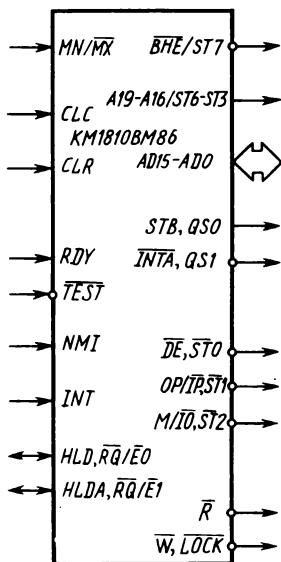


Рис. 1.9. Схема подключения выводов МП KM1810BM86

ных — резервного состояния; имеет низкий уровень при операциях чтения, записи, подтверждения прерывания, в начале циклов обращения к магистрали;

A19—A16/ST6—ST3 — выходы старшего полубайта адреса памяти — состояния; при адресации внешнего устройства на выходах — нулевые значения. После выдачи адреса формируются биты состояния: ST6=0, ST5 — разрешение прерывания (обозначает начало каждого цикла); ST4, ST3 — код сегментного регистра, используемого в данный момент (00 — дополнительный сегмент данных, 01 — сегмент стека; 10 — сегмент команд; 11 — сегмент данных);

AD15—AD0 — двунаправленный мультиплексируемый канал адреса — данных. В начале цикла передаются адреса, а затем канал используется для передачи данных.

Выходы BHE и AD0 используются для управления доступом к банкам памяти и тем самым управляют выборкой слов или байтов (табл. 1. 2.).

Таблица 1.2

$\overline{\text{BHE}}$	АО	Разрядность слов	$\overline{\text{BHE}}$	АО	Разрядность слов
0	0	Все слово	1	0	Байт D7 — D0, четный адрес
0	1	Байт D15 — D8, нечетный адрес	1	1	Нет обращения

Устройство управления и синхронизации управляет работой микропроцессора. Управление выполнением команд осуществляется формированием сигналов в зависимости от кодов операций команд, извлеченных из очереди. Синхронизация взаимной работы микропроцессора и сопрягаемых с ним устройств обеспечивается сигналами, поступающими на выводы:

CLC — вход сигнала синхронизации;

CLR — вход начальной установки процессора. Его длительность должна быть не менее четырех тактов синхроимпульсов;

RDY — вход сигнала готовности. Указывает на то, что адресуемые запоминающие и внешние устройства готовы к обмену. Позволяет синхронизировать работу процессора с запоминающими и внешними устройствами;

TEST — вход сигнала программной проверки готовности других устройств в системе. Состояние готовности

опрашивается командой WAIT. Если $\overline{\text{TEST}}=0$, то процессор продолжает обработку, если $\overline{\text{TEST}}=1$, то процессор переходит в состояние ожидания — продолжает опрашивать вывод TEST с интервалом в 5 тактов.

Управление прерываниями осуществляется сигналами, поступающими на входы:

MMI — немаскируемый запрос прерывания, обрабатывается микропроцессором в конце цикла выполнения текущей команды;

INT — маскируемый запрос прерывания. Запоминается внутренним триггером и обрабатывается в последнем такте команды. Маскируемые запросы должны быть разрешены признаком разрешения прерывания $\text{IF}=1$.

Системная «настройка» микропроцессора на минимальный или максимальный режим работы обеспечивается формированием и подачей сигнала на вход $\text{MN}/\overline{\text{MX}}$ управления режимом работы; 0 — максимальный, 1 — минимальный.

В минимальном режиме для управления доступом к локальной магистрали используются выводы:

HLD — вход сигнала захвата, указывающий на то, что другой процессор захватывает локальную магистраль;

HLDA — выход сигнала подтверждения захвата, выдаваемого в ответ на сигнал HLD. Процессор переходит в состояние захвата. При этом магистрали адреса, данных и управления переводятся в высокоимпедансное состояние. После установления низкого уровня на входе захвата процессор также устанавливает сигнал низкого уровня на выходе подтверждения захвата и управление магистралью берет на себя.

В максимальном режиме эти же выводы используются для запросов локального канала по одной из двух двунаправленных линий $\overline{\text{RQ}}/\text{EI}$ и $\overline{\text{RQ}}/\text{EO}$ — запроса — разрешения доступа к локальной магистрали, причем вывод $\overline{\text{RQ}}/\text{EO}$ имеет более высокий приоритет.

Процесс доступа к магистрали выполняется в определенной последовательности:

- устройство, подключенное к каналу и требующее доступа к общим ресурсам (запоминающее или ввода — вывода), формирует импульсный сигнал (первый) запроса длительностью в один такт синхроимпульсов;

- микропроцессор в конце текущего цикла выдает ответный (второй) импульс, подтверждающий возможность доступа к локальной магистрали. В следующем

также микропроцессор отключается от магистрали, т. е. переводит канал адреса — данных и управления в высокоимпедансное состояние;

— по окончании работы с магистралью устройство выдает на ту же линию импульс (третий), указывающий на окончание захвата канала. В следующем такте микропроцессор берет управление на себя.

Управление магистралью при обменах осуществляется сигналами на выводах, функциональное назначение которых определяется заданным режимом работы микропроцессора. В минимальном режиме процессор формирует сигналы, позволяющие управлять магистралью непосредственно.

\overline{STB} — выход stroba адреса; используется для записи адреса в буферный регистр. Выдается в начале каждого цикла;

\overline{INTA} — выход сигнала подтверждения прерывания, по которому читается вектор внешнего прерывания;

\overline{DE} — трехстабильный выход разрешения передачи данных. Используется для управления магистральными формирователями; выдается в циклах чтения, записи, подтверждения прерывания;

OP/\overline{IP} — трехстабильный выход направления передачи данных ($OP/\overline{IP}=1$ — выдача данных, $OP/\overline{IP}=0$ — прием данных);

M/\overline{IO} — трехстабильный выход признака обращения к ОЗУ ($M/\overline{IO}=1$) или внешнему устройству ($M/\overline{IO}=0$);

\overline{R} — трехстабильный выход сигнала чтения устройств, подключенных к локальной шине;

\overline{W} — трехстабильный выход сигнала записи.

В максимальном режиме выводы имеют следующее назначение:

QSI , QSO — сигналы состояния очереди команд. Указывают состояние очереди в такте, после которого было выполнено действие над очередью; 00 — нет операций, из очереди ничего не выбиралось; 01 — первый байт команды; 10 — очередь пуста, была освобождена командой перехода; 11 — следующий байт команды. Выводы обеспечивают возможность внешнего слежения за очередью и используются для совместной работы с сопроцессором;

$ST0$, $ST1$, $ST2$ — трехстабильные выводы сигналов состояния цикла обмена информацией. Используются для формирования сигналов управления магистралью (рис. 1.9): 000 — подтверждение прерывания; 100 — чтение внешнего устройства (VU); 010 — запись в VU ;

110 — останов; 001 — выборка команды; 101 — чтение памяти; 011 — запись в память; 111 — пассивное состояние ($\overline{ST2}$ — логически эквивалентно выходу M/\overline{IO} , $\overline{ST1}$ — выходу OP/\overline{IP});

\overline{LOCK} — трехстабильный выход блокировки системной магистрали. Формируется командой — префиксом \overline{LOCK} , располагаемой перед командой, после которой осуществляется доступ к системной магистрали. Действует до конца выполнения команды и запрещает доступ к системной магистрали другим процессорам.

Управление работой микропроцессора требует формирования сигналов, обеспечивающих реализацию начальной установки микропроцессора, последовательного выполнения команд, синхронизации с внешними устройствами, обработки прерываний, захвата магистралей и останова микропроцессора.

Начальная установка и запуск микропроцессора КМ1810ВМ86 осуществляются подачей сигнала единичного уровня на вход CLR длительностью не менее 50 мкс после достижения номинального значения напряжения питания. Для установки в исходное состояние в процессе работы (перезапуск) необходимо на вход CLR подать сигнал единичного уровня длительностью не менее четырех периодов тактовой частоты.

При появлении сигнала на входе CLR микропроцессор прекращает внутренние операции, канал адреса — данных переводит в высокоимпедансное состояние, а выходы управления — в состояния, указанные ниже:

A19/ $\overline{ST6}$ — A16/ $\overline{ST3}$	Высокоимпедансное (третье)
AD 15 — AD 0	»
$\overline{BHE}/ST 7$	Переход через 1 в третье
M/\overline{IO} ($\overline{ST 2}$)	То же
OP/\overline{IP} ($\overline{ST 1}$)	»
\overline{DE} ($\overline{ST0}$)	»
\overline{W} (\overline{LOCK})	»
\overline{R}	»
\overline{INTA}	1
STB	0
HLDA	0
$\overline{RQ}/\overline{EO}$	1
$\overline{RQ}/\overline{EI}$	1
QSO	0
QSI	0

Переход в начальное состояние осуществляется в соответствии с рис. 1.10. При этом регистры процессора устанавливаются в состояния, указанные ниже:

Регистры	Содержимое
Флагов	0000
Указателя команд IP	0000
Сегмента команд CS	FFFF
Сегмента данных DS	0000
Сегмента стека SS	0000
Сегмента ES	0000
Очереди	пусты

Так как содержимое регистра CS равно FFFF, регистра адреса команды IP — 0000, то на выходах микропроцессора устанавливается физический адрес памяти FFFF. По этому адресу микропроцессор автоматически выбирает первую команду инициализации, обычно команду прямого межсегментного перехода JMP на начало системной программы. Кроме того, поскольку флажок IF разрешения маскируемых прерываний сброшен, необходимо осуществить его программную установку в требуемом месте программы.

При выполнении команд работу процессора можно представить последовательностью циклов обмена, в течение которых микропроцессор обращается к памяти за командами или обменивается данными с памятью или внешними устройствами.

Процессор осуществляет обмен через магистраль адреса — данных, используя ее в режиме разделения вре-

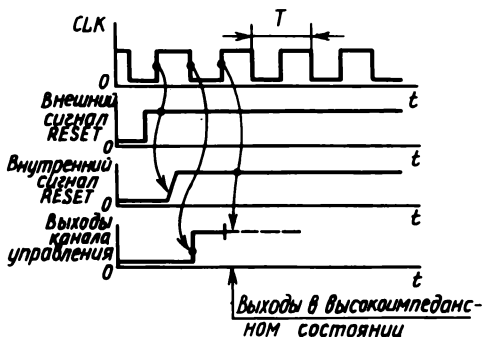


Рис. 1.10. Временные диаграммы перехода в начальное состояние

мени. Каждый цикл состоит минимум из четырех тактов: T1, T2, T3, T4. В такте T1 выдается адрес, в тактах T3, T4 осуществляется передача данных, такт T2 используется в основном для коммутации направления передачи.

Если на входе RDY действует сигнал неготовности ($RDY = 0$), то между тактами T3, T4 выполняются такты ожидания T_w , число которых определяется длительностью сигнала неготовности.

Цикл выполняется, когда заполнена очередь команд или осуществляется обмен данными в процессе выполнения команды. Если цикл обмена не требуется, то между циклами формируются холостые такты T5, число которых зависит от типа выполняемой команды.

В минимальном режиме временные диаграммы цикла обмена показаны на рис. 1.11.

Цикл начинается с формирования в такте T1 сигнала M/\overline{IO} , определяющего тип устройства (ЗУ или УВВ), с которым должен произойти обмен данными. Длительность сигнала M/\overline{IO} равна длительности цикла обмена. Сигнал M/\overline{IO} используется при селекции адреса устройства, с которыми осуществляется обмен данными.

На мультиплексный канал адреса — данных в такте T1 и начале такта T2 микропроцессор выдает адрес ЗУ по разрядам A19—A16, AD15—AD0 или адрес устройства ввода — вывода (УВВ) по разрядам AD15—AD0, а также сигнал \overline{BHE} , указывающий на разрешение передачи старшего байта данных. Сигналы \overline{BHE} и \overline{AO} определяют обмен словом или одним байтом.

В такте T1 микропроцессор выдает сигнал STB единичного уровня, по спаду которого формируется адрес в регистрах. При чтении канал переходит в высокоимпедансное состояние входа, при записи — выдает данные. На выходах регистров адрес остается действительным в течение всего цикла обмена.

По разрядам A19/ST6 — A16/ST3 выдаются сигналы состояния ST6—ST3 с такта T2 по такт T4. Сигнал \overline{R} низкого уровня в тактах T2—T4 цикла чтения указывает адресуемому устройству на необходимость выдачи данных. При отсутствии готовности устройство формирует сигнал READY не менее чем за 35 нс до начала такта T3. Между тактами T3 и T4 микропроцессор находится в состоянии ожидания (такты T_w). О готовности данных устройство сигнализирует появлением сигнала высокого уровня на входе RDY. Адресуемое устройство выдает данные на свободный канал, микропроцессор принимает

ных микропроцессор формирует сигнал \overline{DE} — разрешение передачи данных при готовности данных.

Направление передачи данных через буферы определяет сигнал OP/IP (в цикле чтения — низкий уровень), действующий в течение цикла чтения. По окончании сигнала \overline{R} мультиплексный канал переводится в высокоимпедансное состояние, буферы данных отключаются от канала, освобождая его.

Цикл записи начинается с формирования в такте $T1$ сигнала $M1/\overline{IO}$, указывающего на запись в $ЗУ$ или $УВВ$.

Во время выдачи адреса на магистраль по разрядам $A19-A16$, $AD15-AD0$ (запись в $ЗУ$) или по разрядам $AD15-AD0$ (запись в $УВВ$) в такте $T1$ микропроцессор выдает сигнал разрешения фиксации адреса STB . В такте $T2$ адрес на магистрали заменяется действительными данными, подлежащими записи в адресуемое устройство. Данные остаются действительными до середины такта $T4$ и сопровождаются сигналом \overline{W} нулевого уровня. Сигнал \overline{W} используется в качестве stroba для записи данных в адресуемое устройство.

В такте $T1$ начинается формирование сигнала \overline{DE} на протяжении низкого уровня, который используется для подготовки буферов к передаче данных.

При несоответствии быстродействия адресуемого устройства и микропроцессора вводятся такты ожидания T_w снятием сигнала RDY на время, необходимое устройству для приема данных с канала. При установке сигнала RDY микропроцессор завершает цикл записи в такте $T4$: данные записаны в устройство и сняты с канала; канал переведен в высокоимпедансное состояние; сигналы \overline{W} и \overline{DE} , M/\overline{IO} , $ST7-ST3$ неактивны.

В циклах записи буферы данных должны обеспечить передачу действительных данных с разрядов $AD15-AD0$ канала в адресуемое устройство. Эту функцию выполняет сигнал \overline{DE} , открывающий буфер до появления действительных данных на канале и закрывающий в момент снятия данных с канала или после снятия их.

Сигнал OP/IP в цикле записи имеет высокий уровень.

В максимальном режиме (рис. 1.12) процессор выдает код состояния, а для формирования управляющих сигналов требуется специальный контроллер магистрали. Код состояния выдается по фронту синхроимпульса в такте $T4$ предыдущего цикла или в холостом такте $T5$. По спаду синхроимпульсов контроллер опраши-

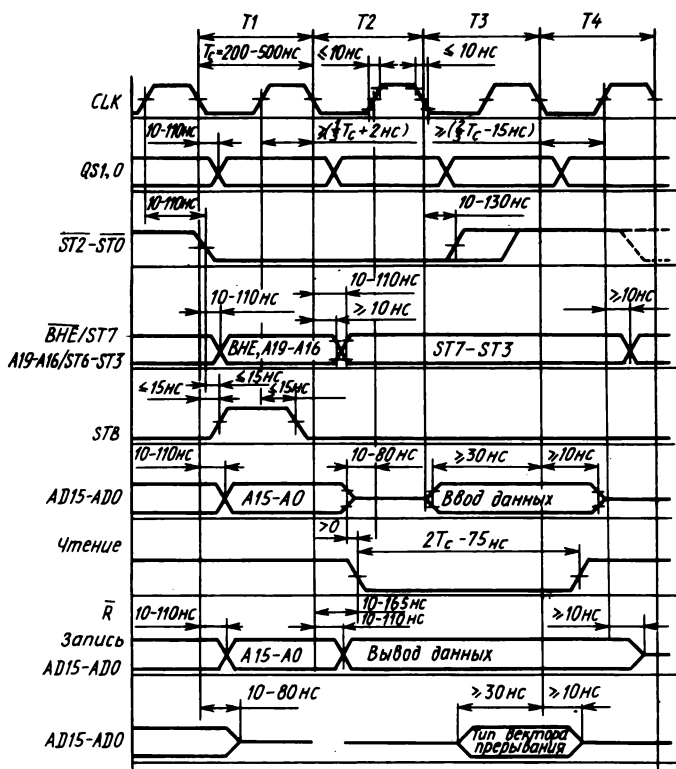


Рис. 1.12. Временные диаграммы работы микросхемы KM1810BM86 в максимальном режиме

вает сигнал состояния, в такте T1 формирует адресный строб, а в следующих тактах — сигналы управления чтением — записью. Управляющие действия контроллера заканчиваются, когда на выходах ST2—ST0 формируется пассивное состояние (111).

Время выполнения команд определяется временем вычисления исполнительного адреса EA при обращениях к памяти и временем исполнения собственно команды. Это время указано в таблицах команд. Число тактов для формирования исполнительного адреса указано ниже:

EA = DISP	— 6
EA = (BX), (BP), (SI) или (DI)	— 5
EA = DISP + (BX), (BP), (SI) или (DI)	— 9

$EA = (BP) + (DI) \text{ или } (BX) + (SI)$	— 7
$EA = (BR) + (SI) \text{ или } (BX) + (DI)$	— 8
$EA = (BP) + (DI) + DISP \text{ или } (BX) + (SI) + DISP$	— 11
$EA = (BP) + (SI) + DISP \text{ или } (BX) + (DI) + DISP$	— 12

Процессор КМ1810ВМ86 осуществляет программную синхронизацию с внешними устройствами программным опросом состояния входа \overline{TEST} . Для этого используется команда ожидания WAIT. Если на входе \overline{TEST} во время выполнения команды присутствует высокий уровень, то процессор находится в состоянии ожидания. Во время программного ожидания возможно прерывание работы процессора. При этом процессор заканчивает исполнение команды WAIT, переходит к обслуживанию запроса прерывания, а после выхода из подпрограммы возвращается к выполнению команды WAIT. Процессор выходит из состояния ожидания, если на входе \overline{TEST} появляется сигнал низкого уровня и действует минимум пять тактов синхроимпульсов.

После восприятия на входе INT сигнала внешнего прерывания микропроцессор переходит в режим подтверждения прерывания, состоящего из двух циклов, разделенных двумя холостыми тактами.

При этом микропроцессор адрес не выдает, но формирует строб STB (рис. 1.11, 1.12). Первый цикл обеспечивает подготовку к приему байта вектора прерывания. Информация в этом цикле не принимается. Во втором цикле читается вектор прерывания для размещения на младшей половине магистрали данных. Чтение вектора прерывания подобно чтению магистрали и осуществляется по сигналу подтверждения прерываний, выдаваемому на выход \overline{INTA} в минимальном режиме или формируемого контроллером в максимальном режиме. При этом формируются также сигналы OP/\overline{IP} и \overline{DE} . Сигнал на выходе M/\overline{IO} имеет низкий уровень.

Для предотвращения захвата магистрали сигналами на входах HLD или $\overline{RQ}/\overline{E}$, начиная с такта T2 первого цикла и кончая тактом T2 второго цикла, формируется внутренний сигнал блокировки LOCK.

Принятый код вектора прерывания микропроцессор умножает на 4, формируя таким образом адресное смещение указателя подпрограммы прерывания в таблице векторов прерывания. Вектор, состоящий из сегментного адреса и адреса указателя команды, читается процессо-

ром. Содержимое регистров сегмента команд CS, указателя команд IP и флажков F загружается в стек, затем в них записываются новые значения прочитанного вектора адреса. Триггеры разрешения прерывания IF и пошагового режима TF сбрасываются. При обращении к таблице векторных адресов сегментные регистры не используются. При формировании адреса адресное смещение складывается с нулем, а биты состояния ST4, ST3 = 10 указывают на то, что ни один сегментный регистр не используется. При подтверждении внешних прерываний реализуется такая последовательность действий:

- формируются два цикла подтверждения прерывания;

- читается новое значение указателя команд IP из таблицы векторов прерываний;

- читается новое значение сегмента команд CS из таблицы векторов прерываний;

- записывается в стек содержимое флажков F;

- записывается в стек предыдущее содержимое регистра CS;

- выбирается код операции первой команды подпрограммы обслуживания прерывания;

- записывается в стек предыдущее значение регистра IP.

После записи в стек содержимого регистра IP процессор действует по заполнению очереди команд и обмену, как при выполнении программы. Число тактов от момента прерывания до начала выполнения подпрограммы составляет 61 такт.

Последовательность действий при программных прерываниях отличается только отсутствием циклов подтверждения прерывания, что приводит к уменьшению числа требуемых тактов до 51 для команд INTn и прерываний пошагового режима, до 52 тактов — для команды INT3, до 53 тактов — для команды INT0.

Захват магистрали в минимальном режиме происходит по наличию сигнала на входе HLD. Временные диаграммы захвата магистрали в минимальном режиме представлены на рис. 1.13.

При поступлении запроса УСК микропроцессор с середины такта T4 текущего цикла обмена или холостого такта формирует на выходе HLDA сигнал подтверждения захвата. Мультиплексный канал адреса — данных и управляющие выводы устанавливаются в высокоимпедансное состояние — микропроцессор отключается от

канала, предоставляя его в пользование устройству, выдавшему запрос захвата. По окончании захвата устройство снимает запрос HLD (устанавливается $HLD=0$), микропроцессор прекращает выдачу сигнала HLDA и продолжает работу с магистралью.

В максимальном режиме доступ может осуществляться к локальной и системным магистралям.

Цикл доступа к локальной магистрали в максимальном режиме состоит из трех фаз (рис. 1.14): запроса, разрешения доступа и освобождения канала. Доступ инициируется сигналом запроса, поступившим на один из выходов $\overline{RQ}/\overline{E}$ с другого процессора. В ответ на запрос главный процессор в конце текущего цикла канала и в холостом такте выдает сигнал разрешения. Канал переводится в высокоимпедансное состояние и в следующем такте освобождается.

При этом устройство обработки продолжает выполнять команды из очереди до тех пор, пока не появится команда, требующая доступа к каналу, либо пока очередь не опустеет.

По окончании доступа к каналу процессор, запросивший его, выдает сигнал освобождения и в следующем

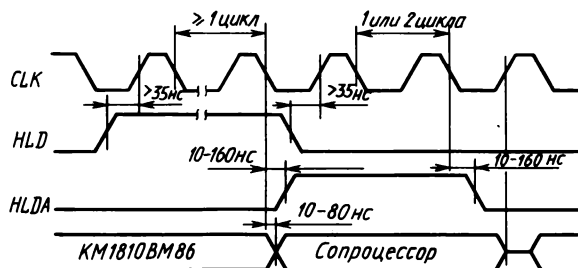


Рис. 1.13. Временные диаграммы захвата магистрали в минимальном режиме

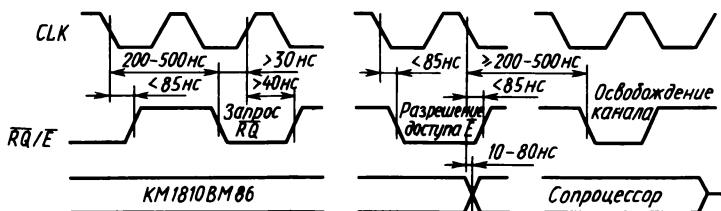


Рис. 1.14. Временные диаграммы захвата локальной магистрали в максимальном режиме

также центральный процессор в свою очередь получает возможность доступа к каналу.

С учетом того, что вход $\overline{RQ}/\overline{EO}$ имеет более высокий приоритет, чем $\overline{RQ}/\overline{EI}$, при одновременном появлении сигналов запроса подтверждение выдается на $\overline{RQ}/\overline{EO}$, а запрос по $\overline{RQ}/\overline{EI}$ удовлетворяется только по окончании обслуживания запроса $\overline{RQ}/\overline{EO}$. Если же запрос $\overline{RQ}/\overline{EO}$ приходит в процессе обслуживания $\overline{RQ}/\overline{EI}$, то он не обрабатывается до окончания обслуживания запроса $\overline{RQ}/\overline{EI}$.

Захват системной магистрали в максимальном режиме осуществляется по сигналу, формируемому на выводе \overline{LOCK} по префиксу команды LOCK. Сигнал остается в активном состоянии в течение выполнения одной команды и используется для блокировки системной магистрали. Две последовательные команды с префиксом LOCK разделяются незаблокированным промежутком. Префикс LOCK обычно используется с командой XCHG, по которой выполняется чтение — запись семафора при обращении к общесистемной памяти.

В минимальном режиме сигнал \overline{LOCK} не формируется, однако префикс LOCK можно использовать для задержки формирования подтверждения захвата на выходе HLDA в ответ на запрос по входу HLD.

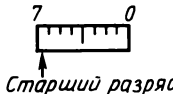
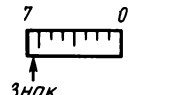
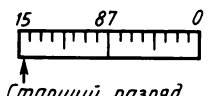
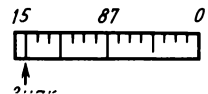
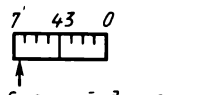
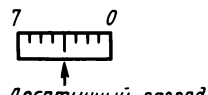
Микропроцессор КМ1810ВМ86 переходит в состояние ОСТАНОВ по команде HLT. При этом в минимальном режиме выдается сигнал на выходе STB, а в максимальном — на выходах $\overline{ST2}$ — $\overline{ST0}$. Во время останова процессор не выдает никаких управляющих сигналов. По появлении сигналов запросов процессор может перейти в состояние ЗАХВАТ, а затем возвратиться в состояние ОСТАНОВ. Выход из состояния ОСТАНОВ осуществляется сигналом начальной установки или сигналом прерывания на входе INT.

1.4. Система команд, программирование микропроцессора

Процессор КМ1810ВМ86 обрабатывает данные четырех типов: двоичные числа без знака; двоичные числа со знаком; упакованные двоично-десятичные числа без знака; распакованные двоично-десятичные числа без знака [6]. Форматы данных представлены в табл. 1.3.

Двоичные числа имеют длину 8 или 16 бит и занима-

Таблица 1.3

Тип данных	Формат	Диапазон
Байт без знака		0—256
Байт со знаком		От — 127 до + 127
Слово без знака		0—65565
Слово со знаком		От — 312768 до 32767
Упакованное десятичное слово		00—99
Распакованное десятичное слово		0—9

ют байт или 2-байтовое слово. Двоично-десятичные числа хранятся в виде байтов — по две десятичные цифры на байт для упакованных двоично-десятичных чисел и по одной десятичной цифре для распакованных двоично-десятичных чисел. Над двоичными числами без знака выполняются арифметические операции сложения, вычитания, умножения и деления. Двоичные числа со знаком представляются в дополнительном коде в формате с запятой, фиксированной после младшего разряда. Над двоичными числами со знаком выполняются операции умножения и деления. Сложение и вычитание выполняется по алгоритмам обработки чисел без знака.

Двоично-десятичные числа представляются в виде чисел без знака. Над упакованными двоично-десятичными числами выполняются операции сложения и вычитания. Результат образуется в двоичном коде, преобразования в двоично-десятичный код осуществляются с помощью операций коррекции. Операции коррекции результатов умножения и деления упакованных двоично-десятичных чисел отсутствуют.

Над двоично-десятичными распакованными числами выполняются операции сложения, вычитания, умножения и деления. При умножении и делении старший полубайт должен быть равен нулю, а при сложении и вычитании может иметь произвольное значение. После выполнения операций сложения, вычитания и умножения осуществляются операции коррекции результатов. При выполнении операций деления коррекция осуществляется перед операцией деления.

Распакованные двоично-десятичные числа представляются аналогично коду ASCII для цифр от 0 до 9. Так как старший полубайт в коде ASCII содержит величину 3, то перед выполнением операций умножения, деления необходимо в старший полубайт поместить величину 0. После выполнения арифметических операций старший полубайт содержит 0 и для получения кода ASCII в него необходимо поместить величину 3.

В результате выполнения арифметических операций устанавливаются следующие флажки признаков:

CF — переноса; устанавливается в 1, если при сложении образуется перенос из старшего разряда, а при вычитании — заем в старший разряд. Флажок используется для обнаружения переполнения при обработке чисел без знака, а также при обработке многобайтовых слов;

AF — дополнительного переноса; устанавливается в 1 при появлении переноса и заема из младшего полубайта в старший. Используется для коррекции результатов при обработке двоично-десятичных чисел;

SF — знака результата; устанавливается по содержанию старшего (8-го или 16-го) разряда: 0 — для положительного, 1 — для отрицательного результатов;

ZF — нулевого результата; устанавливается в 1, если результат равен 0;

PF — четности; устанавливается в 1, если число единиц в результате четно, в противном случае сбрасывается;

OF — переполнения результата; устанавливается в 1.

если происходит переполнение разрядной сетки для чисел со знаком.

Память микропроцессора КМ1810ВМ86, доступная программисту, представлена внутренним СОЗУ объемом 14×16 бит, областью адресов ЗУ объемом 1 М байт и областью адресов внешних устройств объемом 64 К байт.

Внутреннее СОЗУ предназначено для хранения данных и адресов, участвующих в выполнении команд программы, и представляет собой массив регистров с произвольным доступом. В составе СОЗУ имеются (рис. 1.15): блок регистров общего назначения (РОН); блок сегментных регистров; регистр-указатель команд и регистр флажков.

Регистры общего назначения используются для хранения данных и адресов и разделены на две группы по четыре регистра в каждой. Группа регистров данных HL может адресоваться как восемь 8-раз-

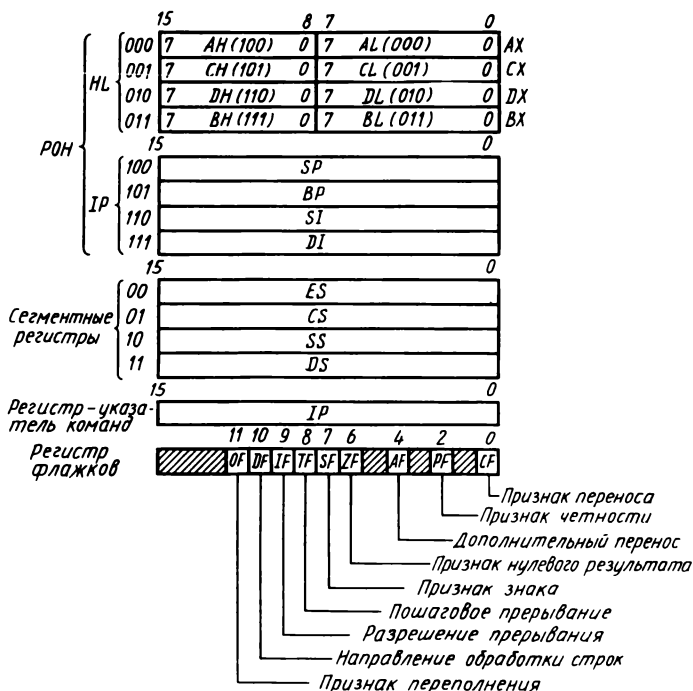


Рис. 1.15. Организация СОЗУ микропроцессора КМ1810ВМ86

рядных регистров AL—BH или как четыре 16-разрядных регистра AX—BX. Группа адресных регистров IP представляет собой два 16-разрядных регистра-указателя адресов и два 16-разрядных индексных регистра. Регистр-указатель стека SP хранит адрес вершины стека относительно его начала; регистр BP — базовые адреса; регистры SI, DI — индекс при адресации массивов данных.

Регистры общего назначения адресуются 3-разрядными двоичными кодами 000—111. Код выбирает один из 8-разрядных регистров (AL—BH) или один из 16-разрядных регистров (AX—DI).

При выполнении некоторых операций РОН имеют специальные функции:

AL — хранение байтов данных при выполнении операций умножения — деления байтов, ввода—вывода байтов, трансляции байтов, операций десятичной арифметики;

AH — хранение байтов данных при выполнении операций умножения — деления байтов;

AX — хранение слов данных при умножении — делении слов, ввод—вывод слов;

BX — адресация таблицы перекодирования при трансляции байтов;

CL — счет числа сдвигов;

CX — счет при выполнении операций обработки строк и циклов;

DX — хранение данных при умножении — делении, хранение указателя адресов при косвенном вводе—выводе;

SP — адресация вершины стека;

SI, DI — адресация строк.

Сегментные регистры предназначены для запоминания начальных адресов сегментов памяти, используемых в текущей программе для хранения команд и данных. Регистр CS указывает сегмент команд, регистр SS — сегмент стека, регистры DS, ES — сегменты данных. Сегментные регистры адресуются 2-разрядным кодом 00—11 или выбираются неявно — по умолчанию.

Регистр-указатель команд IP выполняет функции программного счетчика. Он указывает смещение текущего адреса команды в сегменте команд относительно базового адреса сегмента, хранящегося в регистре CS.

Регистр флажков F предназначен для хранения признаков результатов выполнения операций и

флажков управления режимами прерывания и обработки строк:

TF — флажок пошагового выполнения программ;

IF — флажок разрешения прерывания. Если флажок установлен в 1, то прерывания разрешаются, в противном случае маскируются;

DF — флажок направления обработки строк. Если $DF = 0$, то строки обрабатываются в направлении от младшего адреса к старшему, т. е. адреса в строке автоматически увеличиваются. При $DF = 1$ адреса элементов строки автоматически уменьшаются, т. е. строка обрабатывается от старших разрядов к младшим.

Пространство адресов оперативной памяти представляет собой массив в 1 048 576 байт, адресуемых 20-разрядными адресами 00 000—FFFFF. Двухбайтовые слова располагаются по любым адресам — старший байт хранится в ячейке с большим адресом, младший — с меньшим.

С точки зрения программиста память микропроцессора KM1810BM86 состоит из четырех сегментов объемом 64 К байт (рис. 1.16) — сегментов команд, стека, данных и дополнительного сегмента. Положение каждого сегмента в памяти указывается содержимым соответствующего сегментного регистра — CS, SS, DS или ES. Адрес с разрядностью, равной 20, образуется дополнением 16-разрядного адреса сегмента четырьмя нулевыми младшими разрядами. Таким образом, сегменты могут размещаться в памяти в любом месте общего объема — по

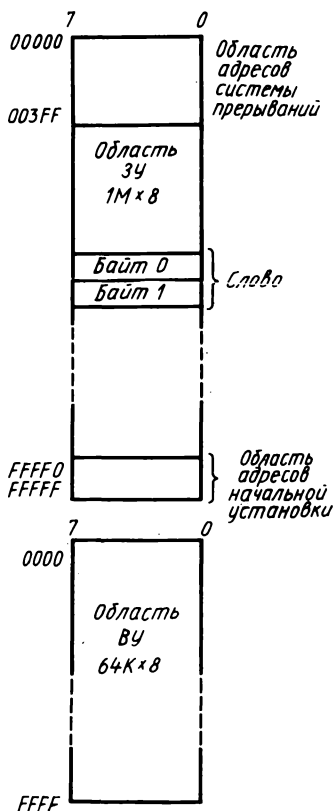


Рис. 1.16. Организация пространства адресов памяти и внешних устройств микропроцессора KM1810BM86 (логическая)

соседству, на расстоянии, частично или полностью перекрываясь. Регистры указывают четыре сегмента, доступные в текущий момент времени. Для получения доступа к другим сегментам содержимое регистров изменяется программным путем.

Две области памяти, расположенные в самых младших адресах 00 000—003FF (1 К байт) и самых старших FFFF0—FFFFF (16 байт), зарезервированы для системы прерываний и начальной установки системы. Физически память микропроцессора организуется в виде двух банков по 512 К байт: старший хранит старшие байты, младший — младшие байты слова. Оба банка адресуются параллельно, а доступ к ним осуществляется сигналами выборки, формируемыми процессором.

Слова могут располагаться в памяти по любым адресам без выравнивания по четным адресам, при этом экономится память. Слова данных, расположенных по четным адресам, выбираются за один цикл обращения к памяти. Если слово располагается по нечетному адресу, то требуется два обращения к памяти — вначале за младшим, а затем за старшим байтом. Поэтому для повышения производительности процессора 16-разрядные слова необходимо располагать по четным адресам — выравнивать данные.

Пространство адресов ввода—вывода микропроцессора составляет массив ячеек объемом 64 К байт. Это пространство не сегментировано и адресуется младшими 16 разрядами 20-разрядного адреса. Физически пространство адресов ввода — вывода организуется как 32 К 16-разрядных слов.

Процессор может обмениваться с внешними устройствами 8- или 16-разрядными словами; 8-разрядные устройства имеют (в зависимости от того, к какой части канала данных они подключены — к младшему или к старшему байту) либо четные, либо нечетные адреса; 16-разрядные устройства должны иметь четные адреса, с тем чтобы осуществлять пересылку за один цикл доступа. Область адресов FFFF8—FFFFF (9 байт) зарезервирована для специальных применений.

Устройства ввода—вывода процессора могут размещаться в пространстве адресов памяти, при этом над их содержимым можно совершать действия, аналогичные действиям над содержимым памяти. Это позволяет повысить гибкость программирования ввода—вывода, однако при этом уменьшается объем доступной памяти.

Процессор КМ1810ВМ86 адресует внутреннее СОЗУ, память и устройства ввода — вывода.

Для адресации РОН внутреннего СОЗУ используется 3-разрядный двоичный код, а для адресации сегментных регистров — 2-разрядный код.

Память адресуется 20-разрядным физическим адресом, а программы оперируют с 16-разрядными логическими адресами. Логически адрес состоит из базы сегмента и смещения. База сегмента представляет собой базовый адрес сегмента и содержится в сегментных регистрах, а адресные смещения определяются способом адресации, указанным в команде, и представляют собой расстояние в байтах от базы сегмента до адресуемого элемента. Оба адреса являются 16-разрядными двоичными числами без знака.

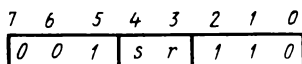
Базовый адрес сегмента указывает начало сегмента, а адрес смещения — расположение элемента сегмента относительно его начала. Физический адрес образуется сложением сдвинутого на четыре разряда влево базового адреса сегмента с адресным смещением. Номер сегментного регистра, участвующего в вычислении физического адреса, не указывается в команде, а выбирается по умолчанию в зависимости от типа адресуемого элемента (табл. 1.4).

Кроме того, выбор сегментов осуществляется специальным байтом префикса, располагаемого перед командой. Префикс в мнемокоде команды записывается перед адресом в форме `sg: адрес`. Коды сегментных регистров:

Таблица 1.4

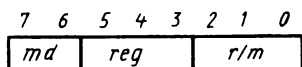
Цель обращения к памяти	Используемый сегментный регистр	Источник смещения
Выборка команд	CS	IP
Операции со стеком	SS	SP
Выборка данных	DS или CS, ES, SS	Исполнительный адрес
Выборка данных при использовании регистра BP в качестве базового	SS или CS, DS, ES	Исполнительный адрес
Адресация строки источника	DS или CS, ES, SS	SI
Адресация строки приемника	ES	DI

ES—00; CS—01; SS—10 и DS—11. Формат префикса имеет вид



Исполнительный адрес ЕА вычисляется на основе содержимого 16-разрядных регистров-указателей ВХ, ВР, SI, DI и смещения DISP, располагаемого в команде и указывающего относительный адрес операнда. Разрядность смещения DISP равна байту или 16-разрядному слову.

Способ вычисления исполнительного адреса ЕА определяется способом адресации, указанным кодом операции команды или специальным байтом. Специальный байт располагается после байта кода операции и называется постбайтом адресации:



Поле reg определяет разрядность (8 или 16) РОН, который содержит операнд команды типа БАЙТ или СЛОВО, указанный в поле W кода операции. Коды и обозначения РОН в зависимости от поля W приведены в табл. 1.5.

Поле md определяет местонахождение и способ адресации операнда. Если md = 00, то операнд находится в памяти и используется косвенная адресация через регистры ВХ, SI, DI; при r/m = 110 используется прямая адресация. Если md = 01, то операнд находится в памяти; используются косвенная адресация через регистры ВХ, ВР, SI, DI и 8-разрядное смещение DISP 8 (знаково расширяемое до 16 разрядов). Если md = 10, то операнд находится в памяти. Используются косвенная адресация через регистры ВХ, ВР, SI, DI и 16-разрядное смещение DISP 16. Если md = 11, то операнд находится в РОН.

Таблица 1.5

Код reg	W = 0	W = 1	Код reg	W = 0	W = 1
000	AL	AX	100	AH	SP
001	CL	CX	101	CH	BP
010	DL	DX	110	DH	CI
011	BL	BX	111	BH	DI

Таблица 1.6

г/м	md = 00	md = 01	md = 10
000	BX+SI	BX + SI + DISP8	BX + SI + DISP16
001	BX + DI	BX + DI + DISP8	BX + DI + DISP16
010	BP + SI	BP + SI + DISP8	BP + SI + DISP16
011	BP + DI	BP + DI + DISP8	BP + DI + DISP16
100	SI	SI + DISP8	SI + DISP16
101	DI	DI + DISP8	DI + DISP16
110	DISP16	BP + DISP8	BP + DISP16
111	BX	BX + DISP8	BX + DISP16

Поле г/м в зависимости от местонахождения операнда определяет код РОН, содержащий операнд команды (при $md = 11$) или способ вычисления исполнительного адреса ЕА (при $md \neq 11$).

Способы вычисления исполнительного адреса ЕА приведены в табл. 1.6.

Регистровая адресация используется для указания регистров, участвующих в выполнении операции. Адрес одного регистра определяется кодом операции. В двухоперандных командах используется постбайт, адресующий два регистра ($md = 11$).

Непосредственная адресация используется в большинстве операций с двумя операндами и задается кодом операций или постбайтом адресации, причем один из операндов располагается непосредственно в команде, второй — в ячейке памяти или в РОН СОЗУ.

В командах с 1-байтовым операндом непосредственный операнд имеет длину 1 байт; в командах с 2-байтовыми словами — занимает 2-байтовое поле команды. Некоторые команды могут иметь непосредственный 1-байтовый операнд. Перед использованием в операции такой операнд расширяется с учетом знака до 16-разрядного слова.

Прямая адресация используется для указания операндов в любом месте сегмента памяти по исполнительному адресу. Если способ прямой адресации указывается в байте кода операции, то адрес располагается за байтом кода операции. Прямая адресация может также указываться постбайтом адресации. При этом $md = 00$, г/м = 110 и прямой адрес равен 16-разрядному смещению DISP 16, расположенному за постбайтом.

Косвенная регистровая адресация используется для адресации памяти по содержимому регистров BX, SI или DI и задается постбайтом адресации при $md = 00$, r/m , равным 100, 110 или 111.

Косвенная регистровая адресация используется при обработке строк с автоматическим увеличением или уменьшением. В качестве регистров-указателей используются регистры SI и DI, неявно адресуемые байтом кода операции, а направление изменения их содержимого задается флажком DF.

Стековая адресация используется для адресации стека. Исполнительный адрес определяется содержимым регистра-указателя стека SP, задаваемого неявно кодом операции. Его содержимое автоматически уменьшается на 2 при занесении слова в стек и увеличивается на 2 после извлечения слова из стека.

Базовая адресация позволяет осуществить доступ к структурам данных. Базу структуры при этом указывает базовый регистр, а расстояние элемента относительно базы — регистр смещения в команде DISP. Базовая адресация задается постбайтом адресации. В качестве базового может использоваться любой регистр — BX, BP, SI или DI. Исполнительный адрес формируется сложением 16-разрядного базового адреса и 8- или 16-разрядного смещения, содержащегося в команде. Если используется смещение DISP 8, то оно является числом со знаком, расположенным в диапазоне от -128 до $+127$.

Индексная адресация логически эквивалентна базовой адресации с 16-разрядным смещением. При этом 16-разрядное смещение представляет собой базисный (относительный) адрес, а базовый регистр содержит значение индекса. Индексная адресация применяется для доступа к элементам массива данных. Смещение задает начало массива, а содержимое индексного регистра — положение элемента относительно начала массива.

Это содержимое изменяется с помощью простых арифметических операций. Регистры BX и BP обычно используются как базовые, а SI и DI — как индексные. При этом регистр BP позволяет адресовать операнды из стекового сегмента. Базово-индексная адресация используется для вычисления адреса операнда, равного сумме базового адреса и смещения операнда относительно базы. Этот способ адресации также задается постбайтом адре-

саций. При этом регистры ВХ и ВР являются базовыми, а SI и DI — индексными.

Базово-индексная адресация применяется для адресации массивов, содержащихся в структуре данных и двумерных массивов. Для адресации массивов, расположенных в стековом сегменте, в качестве базового используется регистр ВР.

Относительная адресация используется для адресации команд в сегменте команд. Исполнительный адрес вычисляется как сумма содержимого указателя команд IP и 8- или 16-разрядного смещения со знаком, находящегося в команде. Содержимое IP при этом указывает адрес байта, следующего за текущей командой. Этот способ адресации задается кодом операции. Применение постбайта адресации для вычисления адреса дает возможность использовать комбинации способа относительной адресации и любого другого, задаваемого постбайтом. В случае непосредственной адресации для определения адресов команд в поле команды указывается 32-разрядное слово, позволяющее изменять адреса в регистрах CS и IP одновременно. Для этой же цели используются и другие способы адресации, задаваемые постбайтом.

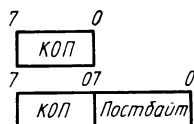
Адресация устройств ввода — вывода, расположенных в пространстве адресов памяти, осуществляется способами адресации памяти.

Обращения к устройствам ввода — вывода, расположенным в области адресов внешних устройств (ВУ), осуществляются с помощью прямой и косвенной регистровой адресации. При прямой адресации адрес ВУ указывается 8-разрядным словом в команде (адресуется 256 устройств ввода — вывода). Косвенная регистровая адресация аналогична косвенной регистровой адресации операндов. Адрес ВУ размещается в регистре DX, что позволяет адресовать 65 536 внешних устройств. Содержимое регистра DX может изменяться в процессе выполнения программы, что позволяет обращаться к группе устройств ввода — вывода в цикле.

Форматы команд микропроцессора КМ1810ВМ86 имеют различные длины — от 1 до 6 байт. В первом байте располагается код операции, последующие байты используются для представления адресной информации. Перед байтом кода операции может располагаться один или несколько 1-байтовых префиксов команды. По числу полей, занятых для представления адресной информации,

форматы команд микропроцессора делятся на безадресные, одно- или двухадресные.

В *безадресных командах* поле адресной информации либо не используется, либо представляется постбайтом способа адресации. Безадресные форматы используются для кодирования операций, выполняемых над содержимым отдельных регистров, битов флажков, а также операций типа регистр—регистр, регистр—стек, регистр—память, регистр—ВУ:



Примечание. Здесь и далее обозначено: $W = 0$ — байт; $W = 1$ — слово; *data* — непосредственные данные; *addr* — адрес ячейки памяти; *port* — адрес ВУ; *disp* — смещение (*l* — младший, *h* — старший байты); *md* — режим 16-разрядного смещения.

Для адресации регистров применяются способы неявной и регистровой адресации, а для адресации памяти и ВУ — косвенной регистровой или стековой адресации. Способы адресации и адреса регистров указываются в поле байта кода операции или в постбайте адресации.

В *командах с одноадресным форматом* в поле байта кода операции или постбайта указывается операнд при непосредственной адресации, адрес — при прямой адресации, смещение — при базовой, индексной, базово-индексной и относительной адресациях. Адресный формат используется при выполнении операции над содержимым ячеек памяти типа: регистр — непосредственный операнд; регистр—память; память—стек; регистр—ВУ. Длина таких команд 2—5 байт в зависимости от способа адресации и типа команды. Способ адресации указывается в коде операции или постбайте. Разрядность слов непосредственных данных указывается в байте кода операции признаком длины слова W . Для размещения в команде только одного байта для 16-разрядного слова используется признак S : если $S = 0$, то 16-разрядное слово указывается в команде; если $S = 1$, то 16-разрядное слово формируется знаковым расширением байта, указываемого в команде:

<i>КОП</i>	<i>W</i>	<i>data</i>	<i>data, W=1</i>
------------	----------	-------------	------------------

<i>КОП</i>	<i>W</i>	<i>reg</i>	<i>data</i>	<i>data, W=1</i>
------------	----------	------------	-------------	------------------

<i>КОП</i>	<i>SW</i>	<i>Посмбайт</i>	<i>data</i>	<i>data, SW=01</i>
------------	-----------	-----------------	-------------	--------------------

<i>КОП</i>	<i>data-l</i>	<i>data-h</i>	<i>data-l</i>	<i>data-h</i>
------------	---------------	---------------	---------------	---------------

<i>КОП</i>	<i>addr-l</i>	<i>addr-h</i>
------------	---------------	---------------

<i>КОП</i>	<i>port</i>
------------	-------------

<i>КОП</i>	<i>Посмбайт</i>	<i>disp</i>	<i>disp, md=10</i>
------------	-----------------	-------------	--------------------

<i>КОП</i>	<i>disp-l</i>	<i>disp-h</i>
------------	---------------	---------------

<i>КОП</i>	<i>disp</i>
------------	-------------

32-разрядное слово непосредственного операнда указывается 4-байтовым полем, располагаемым за байтом кода операции. Такой формат применяется лишь в некоторых командах для одновременной загрузки двух регистров, один из которых сегментный.

Разрядность адреса при использовании прямой адресации составляет 16 бит для адресации памяти или 8 бит для адресации ВУ. При указании прямой адресации кодом операции адрес располагается после байта кода операции, а при использовании постбайта прямой адрес равен 16-разрядному смещению, располагаемому после постбайта.

Разрядность смещения для базового, индексного и базово-индексного способов адресации указывается в постбайте полем *md* (*md* = 00 — смещение равно 0; *md* = 01 — в команде указывается 8-разрядное смещение, для получения 16-разрядного слова используется знаковое расширение; *md* = 10 — в команде указывается 16-разрядное смещение).

Смещение располагается после постбайта адресации. В командах с относительной адресацией смещение можно указывать в байте 2 команды использования постбайта.

Второй адрес в двухоперандных командах с одноадресным форматом указывается так же, как и в безадресном формате.

В командах с двухадресным форматом один из операндов является непосредственным. Формат используется для выполнения операций типа память — исполнительный операнд. Этот формат всегда имеет постбайт адресации, после которого располагается смещение, а затем непосредственный операнд. Длина команды составляет 5, 6 байт:

КОП	W/Постбайт	disp	disp,md=0	data	data,W=1
-----	------------	------	-----------	------	----------

В систему команд процессора КМ1810ВМ86 входят следующие группы: команды пересылок; арифметические команды; команды логические и сдвига; команды обработки строк и управления программой.

Команды первой группы пересылают данные между регистрами, регистрами и памятью, регистрами и внешними устройствами. Группа содержит 14 команд: пересылок байтов или слов, работы со стеком, обмена, перекодировки байта, загрузки адресов и ввода — вывода.

По команде MOV выполняются пересылки между регистрами, регистрами и памятью, а также непосредственная загрузка регистров и ячеек памяти. Команда имеет формат MOV dst, src, где dst — приемник, а src — источник:

MOV	$r/m \leftarrow reg$	1000100W	md reg r/m		
	$reg \leftarrow r/m$	1000101W	md reg r/m		
	$r/m \leftarrow Sr$	10001100	md 0Sr r/m		
	$Sr \leftarrow r/m$	10001110	md 0Sr r/m		
	$A \leftarrow m(addr)$	1010000W	addr	addr	
	$m(addr) \leftarrow A$	1010001W	addr	addr	
	$r/m \leftarrow data$	1100011W	md reg r/m	data	data, W=1
	$reg \leftarrow data$	1011W reg	data	data, W=1	

Примечание:

W = 0	W = 1	
reg = 000 — AI	AX	sr = 00 — ES
001 — CL	CX	01 — CS
010 — DL	DX	10 — SS
111 — BL	BX	11 — DS
100 — AH	SP	
101 — CH	BP	
110 — DH	SI	
111 — BH	DI	

Команды PUSH, POP предназначены для работы со стеком загрузки данных из регистров или памяти в стек и их извлечения из стека. Адрес вершины стека указывается регистрами SS и SP. При загрузке в стек содержимое регистра SP вначале уменьшается на 2, а затем слово данных пересылается в стек. При извлечении из стека вначале читается слово данных, а затем содержимое SP увеличивается на 2.

Для сохранения — восстановления в стеке содержимого регистра флажков используются специальные команды PUSH F и POP F, имеющие формат без операнда:

PUSH	стек \leftarrow r/m	11111111 md 110 r/m
	стек \leftarrow reg	01010 reg
PUSHF	стек \leftarrow SP	0005100
	стек \leftarrow F	10011100
POP	r/m \leftarrow стек	10001111 md 000 r/m
	reg \leftarrow стек	01011 reg
POPF	SP \leftarrow стек	0005111
	F \leftarrow стек	10011101

Команда XCHG производит обмен байтами или словами между двумя любыми регистрами, регистром и ячейкой памяти. Для обмена содержимым аккумулятора с любым 16-разрядным регистром используется команда с 1-байтовым форматом XCHG dst, src:

XCHG	r/m \leftrightarrow reg	1000011W md reg r/m
	A \leftrightarrow reg	10010 reg

Команда XLAT выполняет перекодировку символов заменой байта в аккумуляторе AL байтом из таблицы перекодировки. Начало таблицы указывается содержимым регистра BX, а смещение ее элементов определяется содержимым аккумулятора до замены. Размер таблицы составляет объем до 256 байт. Команда имеет формат XLAT src-table, где src-table — адрес таблицы перекодировки:

XLAT	AL \leftarrow m(BX + AL)	11010111
------	----------------------------	----------

Команда LEA используется для загрузки 16-разрядного РОН исполнительным адресом ячейки памяти, адрес которой вычисляется в соответствии с постбайтом способа адресации (регистровую адресацию при этом использовать нельзя). Команда имеет формат LEA dst, src.

Команды LDS, LES предназначены для начальной загрузки 4-байтового логического адреса в POH и сегментные регистры DS и ES. Логический адрес размещается в соседних ячейках памяти — вначале смещение, а затем база сегмента. Команды имеют формат такой же, как и команда LEA:

LEA	$reg \leftarrow EA$	10001101	md reg r/m
LDS	$DS, reg \leftarrow r/m$	11000101	md reg r/m
LES	$ES, reg \leftarrow r/m$	11000100	md reg r/m

Для одновременной загрузки логических адресов в регистры SP и SS специальной команды нет. Поэтому необходимо выполнять две команды пересылки MOV: MOV SS, <адрес> и MOV SP, <адрес>. До полной загрузки логического адреса прерывания запрещаются.

Команды LAHF и SAHF используются для пересылки флажков SF, ZF, AF, PF и CF в регистр AH и наоборот и предназначены для обеспечения совместимости с системой команд микропроцессора K580 BM80A:

LAHF	$A \leftarrow F$	10011111
SAHF	$F \leftarrow A$	10011110

Команды ввода — вывода IN, OUT представлены двумя форматами. В формате с прямой адресацией адрес порта указывается во втором байте, что позволяет обращаться к устройствам с номерами от 0 до 255. В 1-байтовом формате адрес ВУ указывается регистром DX, при этом число адресуемых устройств равно 65 536. Обмен осуществляется через аккумулятор. Формат команд IN acc, port; OUT acc, port:

IN	$A \leftarrow BU(N)$	1110010W	N
	$A \leftarrow BU(DX)$	1110110W	
OUT	$BU(N) \leftarrow A$	1110011W	N
	$BU(DX) \leftarrow A$	1110111W	

Арифметические команды реализуют операции сложения, сложения с переносом, инкремента, вычитания с заемом, декремента, умножения и деления. Команды сложения ADD, сложения с переносом ADC, вычитания SUB и вычитания с заемом SBB выполняют операции над содержимым регистров процессора или регистров и ячеек памяти, результат помещается в регистры или ячейки памяти (т. е. выполняются операции типа $reg \leftarrow reg \text{ on } r/m$; $r/m \leftarrow r/m \text{ on } reg$).

Способ адресации ячейки памяти задается постбайтом. Смещение при необходимости указывается после постбайта. Разрядность операндов указывается признаком W.

В качестве второго операнда можно использовать также непосредственный операнд ($r/m \leftarrow r/m \text{ on data}$), который при необходимости располагается после постбайта и адресного смещения. Разрядность непосредственного операнда задается признаком W, который позволяет задавать в поле операнда 8-разрядное число, требуемое 16-разрядное число формируется его знаковым расширением. Второй формат команд с непосредственным операндом выполняет операцию только над содержимым аккумулятора, что обеспечивает совместимость с микропроцессором K580.

По результатам выполнения операций сложения — вычитания устанавливаются все признаки: OF, SF, ZF, AF, PF, CF.

Команды сложения — вычитания с переносом — заемом позволяют реализовать сложение — вычитание многобайтовых слов; команда сравнения CMP выполняет ту же операцию, что и команда вычитания SUB, но результат операции не запоминается, фиксируются лишь признаки. Форматы команд имеют вид

		7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0
ADD	$r/m \leftarrow r/m + reg$	0000 000W md reg r/m
	$reg \leftarrow reg + r/m$	0000001W md reg r/m
	$r/m \leftarrow r/m + data$	100000SW md000r/m data data, SW=01
	$A \leftarrow A + data$	0000010W data data, W=1
ADC	$r/m \leftarrow r/m + reg + CF$	0001000W md reg r/m
	$reg \leftarrow reg + r/m + CF$	0001001W md reg r/m
	$r/m \leftarrow r/m + data + CF$	100000SW md010r/m data data, SW=01
	$A \leftarrow A + data + CF$	0001010W data data, W=1
SUB	$r/m \leftarrow r/m - reg$	0010100W md reg r/m
	$reg \leftarrow reg - r/m$	0010101W md reg r/m
	$r/m \leftarrow r/m - data$	100000SW md101r/m data data, SW=01
	$A \leftarrow A - data$	0010110W data data, W=1
SBB	$r/m \leftarrow r/m - reg - CF$	0001100W md reg r/m
	$reg \leftarrow reg - r/m - CF$	0001101W md reg r/m
	$r/m \leftarrow r/m - data - CF$	100000SW md011r/m data data, SW=01
	$A \leftarrow A - data - CF$	0001110W data data, W=1
CMP	$F \leftarrow r/m - reg$	0011100W md reg r/m
	$F \leftarrow reg - r/m$	0011101W md reg r/m
	$F \leftarrow reg - data$	100000SW md011r/m data data, SW=01
	$F \leftarrow A - data$	0011110W data data, W=1

При выполнении операции над двоично-десятичными числами полученный двоичный результат преобразуется

в двоично-десятичные коды командами AAA, AAS для распакованных чисел и командами DAA, DAS для упакованных двоично-десятичных чисел:

AAA	ASCII коррекция	00110111
DAA	2-10 коррекция	00100111
AAS	ASCII коррекция	00111111
DAS	2-10 коррекция	00101111

Команды инкремента — декремента INC, DEC позволяют увеличить или уменьшить на 1 содержимое регистра или ячейки памяти. Команда NEG формирует число с противоположным знаком в дополнительном коде. Команды INC, DEC, NEG имеют формат типа INC dst. При выполнении команд инкремента — декремента изменяются все признаки, кроме CF. В команде NEG флажок CF устанавливается в том случае, если результат не равен нулю. В противном случае он сбрасывается:

INC	$r/m \leftarrow r/m + 1$ $reg \leftarrow reg + 1$	1111111W md 000r/m 01000 reg
DEC	$r/m \leftarrow r/m - 1$ $reg \leftarrow reg - 1$	1111111W md 001r/m 01001 reg
NEG	$r/m \leftarrow -r/m$	1111011W md 011r/m

Команды умножения — деления MUL, DIV выполняют операции над двоичными и двоично-десятичными неупакованными числами без знака, а IMUL, IDIV — над двоичными числами со знаком. При умножении — делении операндов длиной в 1 байт операции выполняются над содержимым аккумулятора AL и операндом из регистра или ячейки памяти. Двухбайтовый результат при умножении размещается в AX. При делении в AL размещается частное, а в AH — остаток. При умножении — делении 2-байтовых слов операция выполняется над содержимым AX и операндом из регистра или ячейки памяти. Результат размещается в регистрах DX, AX (при умножении в DX — старшие разряды, в AX — младшие; при делении в DX — остаток, в AX — частное). Перед делением разрядность делимого должна быть удвоена. Для этого используются команды преобразования байтов в слова CBW и слов в 2-байтовые слова CWD. Эти команды осуществляют знаковое расширение содержимого AL или AX. Результат размещается в AX или в DX

и AX. Формат команд умножения — деления MUL dst имеет вид

MUL	$\{AX\} \leftarrow \{AL\} \times r/m$	<table><tr><td>1111011W</td><td>md110r/m</td></tr><tr><td>1111011W</td><td>md101r/m</td></tr></table>	1111011W	md110r/m	1111011W	md101r/m
1111011W	md110r/m					
1111011W	md101r/m					
IMUL	$\{DX, AX\} \leftarrow \{AX\} \times r/m$					
DIV	$\{AH, AL\} \leftarrow \{AL\} : r/m$	<table><tr><td>1111011W</td><td>md110r/m</td></tr><tr><td>1111011W</td><td>md111r/m</td></tr></table>	1111011W	md110r/m	1111011W	md111r/m
1111011W	md110r/m					
1111011W	md111r/m					
IDIV	$\{DX, AX\} \leftarrow \{AX\} : r/m$					
CBW	$AX \leftarrow \text{sign } AL$	<table><tr><td>10011000</td></tr></table>	10011000			
10011000						
CWD	$DX, AX \leftarrow \text{sign } AX$	<table><tr><td>10011001</td></tr></table>	10011001			
10011001						

Если старшая половина результата умножения для команды MUL равна 0 или не является знаковым расширением младшей части для команды IMUL, то устанавливаются флажки OF и CF, в противном случае они сбрасываются. Состояние флажков SF, ZF, AF, PF не определено.

Если частное превосходит максимальное значение, то, так же как и при попытке деления на 0, вырабатывается прерывание типа 0, частное и остаток не определены. Состояние флажков признаков не определено. Нецелые частные округляются до целых. Остаток деления чисел со знаком имеет тот же знак, что и делимое.

Для коррекции результатов умножения или деления неупакованных двоично-десятичных чисел используются команды AAM и AAD. Команда AAM выполняется после команд умножения, команда AAD — перед командой деления и корректирует делимое в AL. Команды модифицируют флажки SF, ZF, PF. Состояние остальных флажков признаков не определено.

Для получения правильных результатов при умножении и делении распакованных двоично-десятичных чисел необходимо, чтобы старший байт операндов был равен 0:

AAM		<table><tr><td>11010100</td><td>00001010</td></tr></table>	11010100	00001010
11010100	00001010			
AAD	ASCII Коррекция	<table><tr><td>11010101</td><td>00001010</td></tr></table>	11010101	00001010
11010101	00001010			

Группа логических команд предназначена для обработки отдельных битов операндов и реализует булевы функции НЕ, И, ИЛИ, ИСКЛЮЧАЮЩЕЕ ИЛИ. Это команды NOT, AND, OR, XOR. Команда TEST также выполняет логическую функцию И, но при этом результат не фиксируется, а лишь устанавливаются флажки признаков в логических командах: OF и CF всегда сбрасываются, состояние AF не определено, SF, ZF и PF модифицируются по результату. По команде NOT осу-

ществуется инверсия содержимого регистра или ячейки памяти.

При выполнении двухоперандных логических операций один из операндов находится в регистре, а другой — в регистре, ячейке памяти или является непосредственным операндом. Результат размещается в регистре или ячейке памяти. Способ адресации памяти задается постбайтом. Форматы логических команд аналогичны арифметическим командам:

NOT	$r/m \leftarrow \bar{r/m}$	1111 011W	md 010 r/m		
AND	$r/m \leftarrow r/m \cdot \text{reg}$	0010000W	md reg r/m		
	$\text{reg} \leftarrow \text{reg} \cdot r/m$	0010001W	md reg r/m		
	$r/m \leftarrow r/m \cdot \text{data}$	1000000W	md 100 r/m	data	data, W=1
	$A \leftarrow A \cdot \text{data}$	0010010W		data	data, W=1
OR	$r/m \leftarrow r/m \vee \text{reg}$	0000100W	md reg r/m		
	$\text{reg} \leftarrow \text{reg} \vee r/m$	0000101W	md reg r/m		
	$r/m \leftarrow r/m \vee \text{data}$	1000000W	md 001 r/m	data	data, W=1
	$A \leftarrow A \vee \text{data}$	0000110W		data	data, W=1
XOR	$r/m \leftarrow r/m \oplus \text{reg}$	0011000W	md reg r/m		
	$\text{reg} \leftarrow \text{reg} \oplus r/m$	0011001W	md reg r/m		
	$r/m \leftarrow r/m \oplus \text{data}$	1000000W	md 110 r/m	data	data, W=1
	$A \leftarrow A \oplus \text{data}$	0011010W		data	data, W=1
TEST	$F \leftarrow r/m \cdot \text{reg}$	1000010W	md reg r/m		
	$F \leftarrow r/m \cdot \text{data}$	1111011W	md 000 r/m	data	data, W=1
	$F \leftarrow A \cdot \text{data}$	0010100W		data	data, W=1

Команды сдвига позволяют выполнять арифметические, логические и циклические сдвиги с числом сдвигов от 1 до 255. Число сдвигов указывается признаком V: если V=0, то число сдвигов равно 1, если V=1, то число сдвигов указывается регистром-счетчиком CL. Последнее обстоятельство позволяет вычислить число сдвигов в процессе выполнения программы. Операции выполняются над байтами (W=0) или словами (W=1).

Команда SHL/SAL выполняет сдвиг влево (справа байт или слово заполняется нулями). По команде SHR осуществляется логический сдвиг вправо — разряды слева заполняются нулями; по команде SAR — арифметический сдвиг вправо, разряды слева заполняются знаковым разрядом.

Команды ROL и ROR выполняют циклический сдвиг влево и вправо соответственно. Флажок CF при этом

заполняется выдвигаемым разрядом (слева или справа). Команды RCL и RCR также выполняют циклические сдвиги влево и вправо соответственно, но флажок CF участвует в сдвиге как часть слова.

Команды сдвига имеют форматы типа RCR dst, cnt, где cnt — число сдвигов:

SHL/SAL	$r/m \xleftarrow{\quad} r/m$	$110100VW \quad md \ 010r/m$
SHR	$r/m \xrightarrow{\quad} r/m$	$110100VW \quad md \ 011r/m$
SAR	$r/m \xrightarrow{\quad} r/m$	$110100VW \quad md \ 111r/m$
ROL	$C, r/m \bigcirc r/m$	$110100VW \quad md \ 000r/m$
ROR	$C, r/m \bigcirc r/m$	$110100VW \quad md \ 001r/m$
RCL	$C, r/m \bigcirc r/m$	$110100VW \quad md \ 010r/m$
RCR	$C, r/m \bigcirc r/m$	$110100VW \quad md \ 011r/m$

При сдвигах состояние флажков SF, ZF и PF устанавливается обычным путем, состояние флажка AF не определено, CF устанавливается по выдвигаемому разряду. Флажок OF при сдвигах на один разряд устанавливается в 1, если изменяется содержимое старшего разряда, или в 0, если содержимое старшего разряда не изменяется. При многоразрядных сдвигах состояние флажка OF не определено.

Команды обработки строк реализуют пять основных операций, выполняемых над элементами строк (байтами или словами) длиной до 64 К байт; пересылки MOVS, MOVSB, MOVSW, сравнения CMPS и просмотра для поиска слова SCAS, а также для пересылки в аккумулятор LODS или из него STOS. Команды MOVS, MOVSB, MOVSW пересылают элементы исходной строки в результирующую. Строка-результат всегда размещается в дополнительном сегменте, адресуемом регистром ES. Элементы строки адресуются регистром DI. Исходная строка размещается по умолчанию в сегменте DS, а элементы ее адресуются регистром SI. Можно использовать другой сегмент, указав его в префиксе сегмента. По команде CMPS из исходной строки вычитаются элементы результирующей строки и устанавливаются флажки. Команда используется для сравнения элементов строк; по команде SCAS элементы результирующей строки вычитаются из регистра AL при обработке байтов или из регистра AX при обработке слов и изменяется содержимое флажков. Команда позволяет анализировать строки.

Команды LODS и STOS позволяют пересылать байты или слова из исходной строки в регистр AL или AX и из

регистров AL, AX в результирующую строку. Дополнительные команды обработки позволяют выполнять любые операции над элементами строк. Каждой операции может предшествовать 1-байтовый префикс REP, по которому операция многократно повторяется. Требуемое число повторений указывается в регистре CX. Повторное выполнение операций прекращается: по условию равенства нулю CX (REP) для команд пересылок MOVSB, MOVSW, MOVSW; равенства нулю CX и Z=1 (REPE/REPZ) или равенства нулю CX и Z=0 (REPNE REPNZ) для команд сравнения CMPS и сканирования SCAS. Таким образом, для выполнения операций над строками используются регистры: SI — индекс исходной строки; DI — индекс результирующей строки; CX — счетчик повторений; AL/AX — величина сканирования строк, приемник для LODS, источник для STOS; DF — флажок уменьшения — увеличения; 0 — автоинкрементация, 1 — автодекрементация.

Для начальной установки адресов можно использовать команды LDC, LEC, LEA.

Содержимое регистров SI, DI автоматически модифицируется перед обработкой текущего элемента: при обработке байтов изменяется на ± 1 , при обработке слов — на ± 2 .

Если перед командой записан префикс REP, то после каждого выполнения операции содержимое регистра CX уменьшается на 1. Формат команд обработки строк: MOVSB, CMPS типа MOV dst-str, src-str; команд MOVSB, MOVSW, REP — безоперандные; SCAS, STOS — типа SCAS dst-str, LODS—LODS src-str. Здесь src-str — исходная строка, а dst-str — результирующая:

MOVSB	(DI) ← (SI)	1010010W
CMPS	F(DD) ← (SI)	1010011W
SCAS	F(A) ← (DI)	1010111W
LODS	A ← (SI)	1010110W
STOS	(DI) ← A	1010101W
REP	Повторять, пока CX ≠ 0	1111001Z

Группу команд управления составляют команды передачи управления и команды управления процессором.

Команды передачи управления имеют в своем составе команды безусловного перехода JMP, условных переходов JCC, вызова подпрограмм CALL и

возврата из подпрограмм RET, управления циклами LOOP, LOOPE/LOOPZ, LOOPNE/LOOPNZ, JCXZ и программного прерывания INT — всего 12 команд. Команды передачи управления работают с регистрами указателя сегмента команд CS и указателя команд в сегменте IP. При выполнении команд передачи управления изменяется содержимое регистра IP и CS. Очередь команд при этом аннулируется и программа продолжается с нового адреса.

Команды безусловной передачи управления могут передавать управление команде в текущем сегменте команд NEAR (внутрисегментная передача) или в другом сегменте команд FAR (межсегментная передача).

Адресная информация указывается непосредственно в команде или в постбайте. В первом случае используется относительная для внутрисегментного перехода или непосредственная для межсегментного перехода адресация, во втором — любой способ адресации. Для перехода в текущем сегменте в адресной части команды указывается 16-разрядное смещение (ассемблерное имя NEAR/PTR), представляющее собой двоичное число со знаком. Это число прибавляется к содержимому IP, и таким образом осуществляется переход в диапазоне адресов от $-32\,768$ до $+32\,767$. Если необходимо выполнить переход от -128 до $+127$, то используется формат команды с 1-байтовым смещением (ассемблерное имя NTR). Регистр IP перед сложением содержит адрес следующей команды.

При использовании для перехода в текущем сегменте формата с постбайтом адресации (ассемблерное имя WORD PTR) адрес перехода вычисляется по содержимому регистра или ячейки памяти, указываемых постбайтом, и загружается в регистр IP. Для прямого перехода в другой сегмент (ассемблерное имя FAR PTR) в поле адреса команды указывается 32-разрядный непосредственный адрес, представляющий собой два 16-разрядных слова.

Первое из них загружается в регистр IP, второе — в регистр CS. При косвенном переходе в другой сегмент (ассемблерное имя DWORD PTR) адрес перехода вычисляется с использованием любого, кроме регистрового, способа адресации, указываемого постбайтом. Смещение, требуемое в вычислениях, размещается за постбайтом. Формат команды следующий:

JMP	$IP \leftarrow IP + disp(16)$	11101001	disp-l	disp-h
	$IP \leftarrow IP + disp(8)$	11101011	disp	
	$IP \leftarrow IP + r/m$	11111111	md100r/m	
	$IP, CS \leftarrow IP, CS$	11101010	IP-l	IP-h
			CS-l	CS-h
	$IP, CS \leftarrow r/m$	11111111	md101r/m	

Команды вызова подпрограммы CALL выполняются аналогично команде JMP, за исключением того, что перед переходом в стек загружается адрес возврата. При переходах в текущем сегменте загружается IP, при переходах в другие сегменты в стек загружается также содержимое регистра CS, причем вначале содержимое CS, а затем IP. Тип перехода указывается ассемблерным именем NEAR или FAR, как и для команды JMP.

CALL	$cтек \leftarrow IP, IP \leftarrow IP + disp$	11101000	disp-l	disp-h
	$cтек \leftarrow IP, IP \leftarrow IP + r/m$	11111111	md010r/m	
	$cтек \leftarrow IP, CS$	10011010	IP-l	IP-h
	$IP, CS \leftarrow IP, CS$		CS-l	CS-h
	$cтек \leftarrow IP, CS$	11111111	md011r/m	
	$IP, CS \leftarrow r/m$			

По команде из подпрограммы RET восстанавливается содержимое регистров IP или CS и IP путем извлечения их из стека. Тип команды возврата (внутрисегментного или межсегментного) должен соответствовать типу команды вызова подпрограммы. Он определяется теми же именами: NEAR — для внутрисегментного и FAR — для межсегментного возврата. Команда внутрисегментного возврата восстанавливает содержимое регистра IP, а команда межсегментного возврата — регистров CS и IP.

В формате команды RET может быть указана 2-байтовая величина, которая добавляется к содержимому указателя стека SP. Это позволяет удалить из стека параметры, записанные перед выполнением подпрограммы.

Форматы команд возврата RET или RET < число >:

RET	$IP \leftarrow cтек$	11000011		
	$IP \leftarrow cтек, SP \leftarrow SP + data$	11000010	data-l	data-h
	$IS, IP \leftarrow cтек$	11001011		
	$CS, IP \leftarrow cтек$	11001010	data-l	data-h
	$SP \leftarrow SP + data$			

Команды условного перехода JCC осуществляют переходы в программе в зависимости от условия CS,

указываемого в поле команды. Если условие выполняется, то управление передается команде, расположенной в диапазоне адресов от -128 до $+127$. Величина смещения указывается во втором байте команды. Анализируемые командой условия переходов приведены в табл. 1.7.

Формат команды JCC < метка >.

Команды управления переходами в циклах LOOP, JCXР осуществляют условные переходы в зависимости от содержимого регистра CX, который при этом используется в качестве счетчика циклов. Как и в командах условных переходов, в данном случае при выполнении

Таблица 1.7

Тип условия перехода	Мнемокод	Условие перехода		Код
		Логика	Признак	
По флажку	C	Есть перенос	$CF = 1$	0010
	NC	Нет переноса	$CF = 0$	0011
	S	Отрицательный результат	$SF = 1$	1000
	NS	Положительный результат	$SF = 0$	1001
	E/Z	Операнды равны/нулевой результат	$ZF = 1$	0100
	NE/NZ	Операнды не равны/ненулевой результат	$ZF = 0$	0101
	O	Есть переполнение	$OF = 1$	0000
	NO	Нет переполнения	$OF = 0$	0001
	P/PE	Четно	$PF = 1$	1010
	NP/PO	Нечетно	$PF = 0$	1011
По признаку числа со знаком	L/NGE	Меньше/не больше и не равно	$SF \oplus OF = 1$	1100
	NZ/GE	Не меньше/больше или равно	$SF \oplus OF = 0$	1101
	LE/NG	Меньше или равно/не больше	$(SF \oplus OF) \vee ZF = 1$	1110
	NLE/G	Не меньше и не равно/больше	$(SF \oplus OF) \vee ZF = 0$	1111
По признаку числа без знака	B/NAE	Меньше/не больше и не равно	$CF = 1$	0010
	NB/AE	Не меньше/больше или равно	$CF = 0$	0011
	BE/NA	Меньше или равно/не больше	$CF \vee ZF = 1$	0110
	NBE/A	Не меньше и не равно/больше	$CF \vee ZF = 0$	0111

условия управление передается команде, расположенной в диапазоне адресов от -128 до $+127$, задаваемых смещением в байте 2 команды. Условия переходов в командах следующие: LOOP — $(CX) \neq 0$; LOOPE/LOOPZ — $(CX) \neq 0$ и $ZF = 1$; LOOPNE/LOOPNZ — $(CX) \neq 0$ и $ZF = 0$; JCXZ — $(CX) = 0$.

При выполнении команд содержимое счетчика CX уменьшается на 1 и проверяется на равенство 0. Если $CX \neq 0$, то в командах LOOP управление передается в начало цикла, в противном случае осуществляется выход из цикла. Команда JCXZ используется в начале цикла для обхода в тех случаях, когда регистр содержит нулевое значение, т. е. когда не требуется выполнять циклы.

Формат команд аналогичен формату команды условного перехода:

JCC	Усл. переход	0111CCCC	disp
LOOP	Цикл, пока $(CX) \neq 0$	11100010	disp
LOOPZ/LOOPE	Цикл, $(CX) \neq 0, ZF = 1$	11100001	disp
LOOPNZ/LOOPNE	Цикл, $(CX) \neq 0, ZF = 0$	11100000	disp
JCXZ	Цикл, $(CX) = 0$	11100011	disp

Команды программного прерывания INT передают управление подпрограммам обработки прерываний. Команды загружают содержимое регистров F, CS и IP в стек, сбрасывают флажки разрешения прерывания IF и пошагового выполнения TF, управление передается по векторному адресу подпрограммы обслуживания. Векторный адрес указывается в таблице переходов, которая адресует вектором в зависимости от типа прерывания. В 2-байтовой команде INTn тип прерывания указывается байтом 2 команды, а адрес равен $4n$. В команде INT3 (прерывание пошагового выполнения) адрес таблицы равен $0C_{16}$, в команде INT0 (прерывание по переполнению) — 10_{16} , причем команда INT0 выполняется, если установлен флажок переполнения. Содержимое таблицы переходов загружается в регистры CS и IP, при этом осуществляется переход к подпрограмме.

По команде IRET осуществляется возврат из подпрограммы прерывания — восстанавливается из стека содержимое регистров F, CS и IP. Команда используется для возврата из программ обслуживания как программных, так и аппаратных прерываний. Форматы команд:

INTn	{	стек $\leftarrow F, CS, IP$	11001101	n
INT3		$CS, IP \leftarrow m(4n)$	11001100	
INT0			11001110	
IRET		$F, CS, IP \leftarrow \text{стек}$	11001111	

Команды управления процессором управляют функциями процессора; команды CLC, CMC, STC, CLD, STD, CLI, STI — флажками переноса FC, направления FD и разрешения прерываний FI. Другие команды используются для синхронизации с внешними процессами. Команда HLT вызывает останов процессора. По команде WAIT опрашивается состояние входа TEST: если TEST = 1, то процессор находится в состоянии ОЖИДАНИЕ, в котором проводится анализ входа TEST.

Команда ESC является командой, выполняемой со-процессором. Код команды сопроцессора определяется 6-разрядной константой, указанной в поле, отмеченном знаком X. Адресация операнда осуществляется постбайтом. Если операнд находится в памяти, то процессор вычисляет его исполнительный адрес и читает его из памяти. Сопроцессор принимает операнд и осуществляет его обработку по команде ESC. Если операнд находится в регистре, то процессор никаких действий не выполняет.

Команда LOCK представляет собой 1-байтовый префикс, по которому процессор, работающий в максимальном режиме, формирует на выходе LOCK сигнал блокировки магистрали в процессе выполнения следующей команды. Это позволяет организовать доступ к общесистемным ресурсам мультипроцессорной системы.

По команде NOP в процессоре не выполняется никаких операций и осуществляется переход к следующей команде. Команды управления процессором имеют следующие форматы:

CLC	$CF \leftarrow 0$	11111000
CMC	$CF \leftarrow CF$	11110101
STC	$CF \leftarrow 1$	11111001
CLD	$DF \leftarrow 0$	11111100
STD	$DF \leftarrow 1$	11111101
CLI	$IF \leftarrow 0$	11111010
STI	$IF \leftarrow 1$	11111011
HLT	Остановка	11010100
WAIT	Ожидание	11111011
ESC	Внешняя команда	1101XXXX md XXXr/m
LOCK	Блокировка	11110000
NOP		00000000

Рассмотрим систему прерываний микропроцессора КМ1810ВМ86. Микропроцессор обслуживает прерывания трех типов: внутренние, программные и внешние.

К внутренним прерываниям относятся прерывания по ошибке операции деления и пошагового режима. Они не маскируются и обслуживаются после выполнения команды, в которой они возникли.

Программные прерывания вызываются программными командами INTn, INTO, INT3; они не маскируются и обслуживаются после вызвавшей их команды.

Внешние прерывания вызываются внешними запросами. Процессор имеет два входа для внешних запросов — немаскируемого и маскируемого. Маскируемые прерывания обычно поступают от внешнего контроллера прерываний и представляются потенциальным сигналом, который должен действовать на входе до получения сигнала подтверждения. Маскируемые прерывания могут быть запрещены или разрешены флажком разрешения прерываний IF. Немаскируемые запросы прерывания передаются положительным фронтом сигнала запроса и обслуживаются независимо от состояния флажка IF. Воспринимаются запросы внешних прерываний в конце выполнения текущей команды.

После восприятия прерывания содержимое регистра флажков F, регистров сегмента команд CS и указателя команд IP запоминается в стеке и регистр устанавливается в 0. При этом сбрасываются флажки разрешения IF и пошагового выполнения команд TF, маскируемые прерывания и прерывания пошагового режима запрещаются.

Каждый тип прерывания имеет свой двоичный код — вектор прерываний:

0 — прерывание, возникающее при делении, если результат превышает допустимую разрядность;

1 — прерывание пошагового режима, возникающее при установлении флажка TF;

2 — немаскируемое внешнее прерывание;

3 — прерывание, вызываемое командой INT3 и используемое в основном для остановов при отладке программы;

4 — прерывание по переполнению, возникающее по команде INTO при установлении флажка OF.

Типы других программных прерываний INTn задаются байтом 2 команды.

Тип внешних прерываний определяется вектором прерывания и задается источником прерывания. Код типа прерывания для внутренних, программных и внешних немаскируемых прерываний формируется процессором, а для маскируемых внешних прерываний — контроллером. Всего может обслуживаться 256 прерываний. Код типа прерывания используется для вычисления адреса таблицы векторных адресов, в которой хранятся адреса переходов к подпрограммам.

Векторный адрес состоит из адреса сегмента, в котором хранится подпрограмма, и смещения в сегменте; занимает 4 байт. Таблица располагается в начальной области адресов памяти. Адрес каждого входа в таблицу определяется умножением кода типа прерывания на 4. Объем памяти таблицы составляет $256 \times 4 = 1 \text{ К}$ байт.

Адрес сегмента засылается в регистр сегмента команд CS, а смещение — в регистр указателя команд IP, в результате чего осуществляется переход к подпрограмме обслуживания прерывания. Возврат из подпрограммы обслуживания прерываний осуществляется командой возврата из прерываний IRET. При этом три верхних слова стека пересылаются в регистры F, CS и IP и управление передается команде, при которой началось прерывание.

Обслуживание запросов от одновременно действующих источников осуществляется в соответствии с их приоритетами:

Прерывание по ошибке деления (тип 0)	Высший
Прерывание INTn, INTO (тип 3, 4)	
Немаскируемое внешнее прерывание (тип 2)	↓
Маскируемое внешнее прерывание (тип 5—255)	
Прерывание в пошаговом режиме (тип 1)	Низший

При анализе приоритетов необходимо учитывать маскируемость внешних маскируемых прерываний. Это может привести к перераспределению приоритетов. Если, например, одновременно возникают немаскируемые и маскируемые прерывания, то процессор начинает выполнение немаскируемого прерывания, как имеющего высший приоритет, и маскирует внешние прерывания сбросом флажка IF. Если при этом разрешено прерывание пошагового режима, то оно выполняется сразу после перехода к подпрограмме обслуживания немаскируемого прерывания.

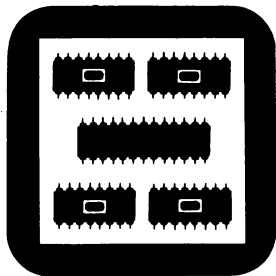
После обслуживания запроса прерывания пошагового

режима управление передается подпрограмме обслуживания немаскируемого прерывания, а запрос к обслуживанию маскируемого прерывания принимается к обслуживанию лишь по окончании этой подпрограммы. Обслуживание маскируемого запроса прерывания можно разрешить во время выполнения этих подпрограмм программным путем, выполнив команду установки флажка разрешения прерывания IF.

Любое немаскируемое прерывание (внешнее, внутреннее или программное) может прервать подпрограмму обслуживания прерывания. Поэтому необходимо избегать случаев появления во время выполнения подпрограммы обслуживания прерываний того же типа. Все подпрограммы должны содержать команды сохранения рабочих регистров процессора в стеке перед их использованием в подпрограмме и команды их восстановления перед окончанием подпрограммы.

Подпрограммы обслуживания программных прерываний часто используются в качестве сервисных программ для обслуживания других программ. Они удобны для использования в системах с динамическим перемещением программ во время их выполнения, так как вызов подпрограмм осуществляется через таблицу с фиксированными адресами, не зависящими от адресов подпрограмм. Адреса подпрограмм могут изменяться путем модификации таблицы указателей векторных адресов.

Проектирование центральных процессоров микроЭВМ «Электроника К1»



Вопросы проектирования центральных процессоров являются одними из важнейших при создании аппаратной основы микроЭВМ.

Широкие функциональные возможности микропроцессоров рассматриваемого семейства обеспечивают построение различных вариантов процессоров. Здесь в структурной взаимосвязи приводятся БИС, входящие в состав центральных процессоров, даются варианты их реализации.

2.1. Проектирование центрального процессора на БИС КР580ВМ80А

Для построения центрального процессора (ЦП) на основе БИС КР580ВМ80А дополнительно требуются микро-схемы формирования сигналов синхронизации и сопряжения с системной магистралью. Структура ЦП, показанная на рис. 2.1, состоит из микропроцессора КР580ВМ80А, схемы синхронизации и схем сопряжения и управления системной магистралью [2, 3, 5].

Для расширения функциональных возможностей ЦП в его состав необходимо дополнительно включить внутреннее (резидентное) ОЗУ; средства обработки запросов магистрали (арбитр магистрали); средства приоритетной обработки запросов прерываний; средства формирования времязадающих сигналов.

Схема синхронизации предназначена для формирования двух последовательностей тактовых импульсов с требуемыми временными и амплитудными параметрами, подаваемыми на входы F1, F2 микропроцессора КР580ВМ80А, а также для синхронизации внешних цепей путем формирования синхроимпульсов фазовых последовательностей ТТЛ-уровней F1_{ТТЛ}, F2_{ТТЛ}; формирования и синхронизации внешних сигналов готовности и сброса

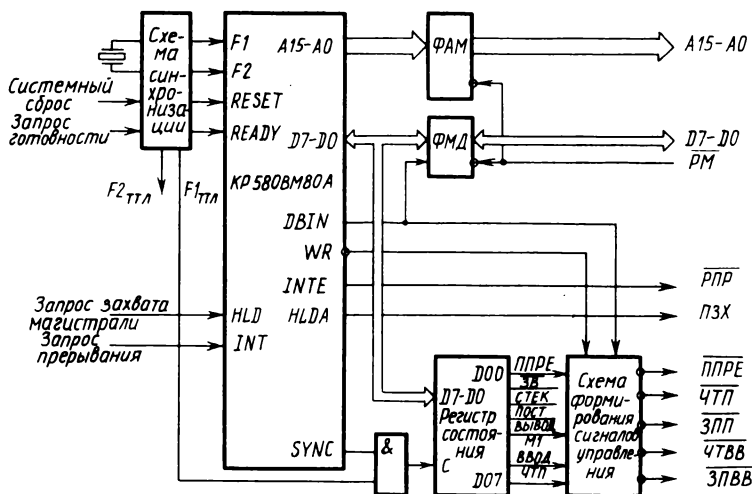


Рис. 2.1. Структура центрального процессора на БИС КР580ВМ80А

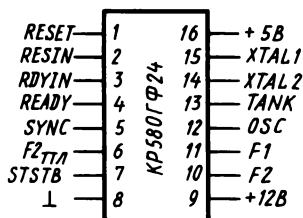


Рис. 2.2. Цоколевка микросхемы КР580ГФ24

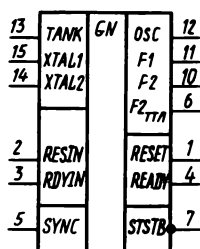


Рис. 2.3. Условное обозначение микросхемы КР580ГФ24

са; управления пошаговой (по командам или циклам) работой микропроцессора.

Для реализации указанных функций в состав микропроцессорного комплекта входит специализированная интегральная схема системного генератора КР580ГФ24 (рис. 2.2 и 2.3).

Микросхема (рис. 2.4) содержит задающий генератор (ЗГ) — стабилизируемый внешним кварцевым резонатором с максимальной частотой 18 МГц; формирователь синхроимпульсов (ФСИ) F1, F2, F2_{ТТЛ}; триггеры (Т) синхронизации сигналов начальной установки и готовности, а также схему формирования строба состояния STSTB,

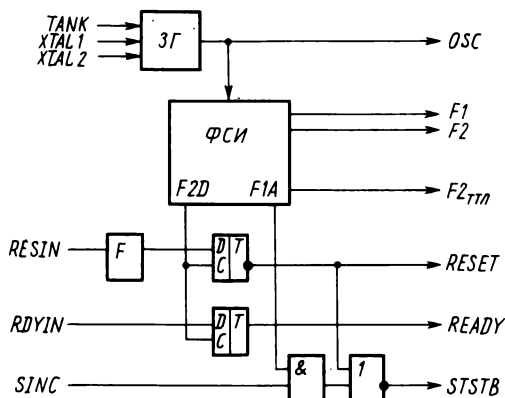


Рис. 2.4. Функциональная схема КР580ГФ24

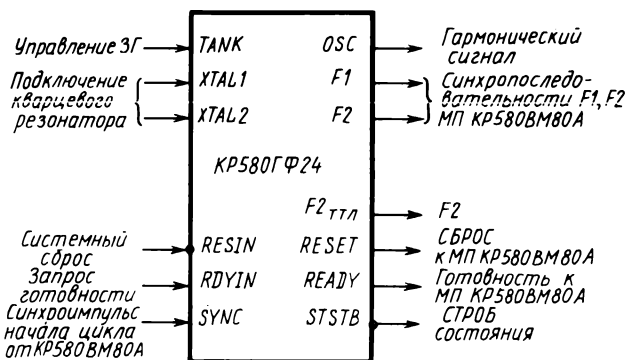


Рис. 2.5. Схема подключения КР580ГФ24

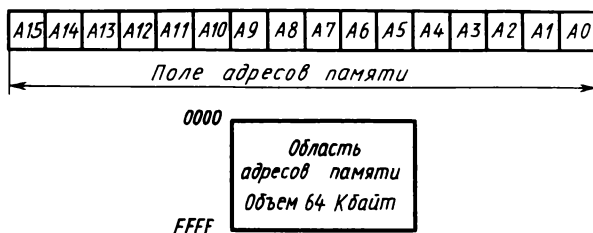
используемого для записи байта состояния в регистр состояния. Сопряжение микросхемы с микропроцессором КР580ВМ80А показано на рис. 2.5.

Цепи сопряжения (интерфейс ЦП) с системой магистралью реализуют следующие основные функции: формирование сигналов адресов памяти и устройств ввода — вывода, поступающих на адресную магистраль, прием и выдачу данных на двунаправленную магистраль данных; формирование управляющих сигналов на магистраль управления.

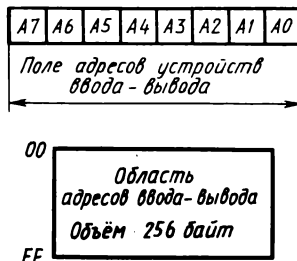
Конкретная структура интерфейса ЦП зависит от способа организации управления ЗУ и ВУ и от структуры

системной магистрали. Память и интерфейс устройства ввода — вывода по отношению к ЦП можно рассматривать как совокупность запоминающих ячеек с произвольным доступом. Микропроцессор КР580ВМ80А выдает адреса памяти и устройств ввода — вывода на общую адресную магистраль. Поэтому важнейшей задачей интерфейса памяти и ввода — вывода является селекция адресов памяти и устройств ввода — вывода. Способ селекции зависит от принятого способа адресации.

Система команд микропроцессора КР580ВМ80А содержит команды ввода — вывода, что позволяет раздельно адресовать память и устройства ввода — вывода. Формат адресного слова и распределение поля адресов памяти имеет вид



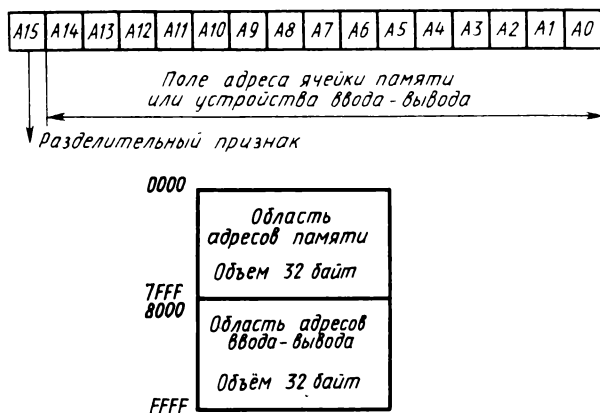
В свою очередь, формат адресного слова и поле адресов ввода — вывода имеет вид



Способ адресации с раздельным полем адресов позволяет адресовать память объемом 64 К байт и 256 устройств ввода — вывода. Память и устройства ввода — вывода при этом разделяются сигналами управления памятью $\overline{ЧТП}$, $\overline{ЗПП}$, устройствами ввода — вывода и $\overline{ЧТВВ}$, $\overline{ЗПВВ}$, формируемыми при выполнении команд обращения к памяти и команд ввода — вывода.

Другой подход состоит в том, что устройства ввода —

вывода рассматриваются как ячейки памяти в общем поле адресов:



Разделительным признаком является один или группа разрядов адреса, чаще всего A15. Если A15=0, то адресуется память; если A15=1, то адресуются устройства ввода — вывода. При таком способе адресации с общим полем адресов объем адресуемой памяти и устройств ввода — вывода составляет 32 К байт каждый. Отличительной особенностью такого способа является возможность использования всех команд обращения к памяти для работы с устройствами ввода — вывода. Например, по команде MOV D, S можно ввести содержимое порта, косвенно адресуемого парой H, L в любой регистр микропроцессора. Используя команду MVIM, D8, можно записать непосредственные данные в порт, а используя команды LHLD ADR и SHLD ADR, можно ввести или вывести 16-разрядное слово. Кроме того, имеется возможность выполнения арифметических и логических операций над содержимым портов ввода — вывода без предварительного занесения их содержимого в аккумулятор. Адресация с общим полем адресов позволяет повысить гибкость и производительность системы.

Аппаратные средства интерфейса ЦП с системной магистралью реализуются магистральными формирователями, регистром состояний и схемой формирования сигналов управления (см. рис. 2.1).

Магистральные формирователи (МФ) предназначены для согласования магистралей ЦП с нагрузкой — адрес-

ными и информационными цепями модулей памяти и ввода — вывода.

Адресная магистраль МП КР580ВМ80А характеризуется следующими электрическими параметрами: напряжение логического 0 не более 0,45 В; напряжение логической 1 не менее 3,7 В; ток нагрузки 1,9 мА.

Магистраль данных процессора D7—D0 характеризуется двумя параметрами, наиболее важными при проектировании систем: минимальным уровнем напряжения входного сигнала (не менее 3,3 В); максимальной нагрузочной способностью (входной ток не более 1,7 мА). С учетом этих параметров выбирают тип МФ. В большинстве случаев используют интегральные схемы К589АП16, К589АП26 или КР580ВА86, КР580ВА87.

Микросхемы К589АП16 (К589АП26) представляют собой 4-разрядные формирователи-коммутаторы.

Микросхемы (рис. 2.6—2.9) имеют: 4-разрядный входной канал D13—D10; 4-разрядный выходной канал D03—D00; 4-разрядный двунаправленный канал DB3—DB0; два 4-разрядных формирователя с трехстабильными выходами; логические вентили управления режимами работы и направления передачи данных. В микросхеме К589АП26 передача осуществляется с инверсией, в микросхеме К589АП16 — без нее.

Формирователи обеспечивают передачу информации при сигнале 0 на выходе выборки CS; при CS = 1 все выходы находятся в высокоимпедансном состоянии. При DCE = 0 информация передается с входов DI на выходы DB, при DCE=1 — с входов DB на выходы DO.

Микросхемы обеспечивают ток нагрузки 50 мА и напряжение логической 1 не менее 3,5 В.

При использовании схем К589АП16 в качестве формирователя магистрали данных (рис. 2.10) управление направлением передачи данных осуществляется сигналом с выхода DBIN микропроцессора КР580ВМ80А. Для отключения магистрали в режиме ПДП ко входу CS подключен выход сигнала системного разрешения магистрали \overline{PM} . В простых системах в качестве этого сигнала используется сигнал HLDA микропроцессора КР580ВМ80А. В сложных системах этот сигнал формирует контроллер ПДП.

Формирователь адресной магистрали (рис. 2.11) выполняет функции адресного буфера и предназначен для увеличения нагрузочной способности. Так как микросхема работает на передачу в одном направлении, ее

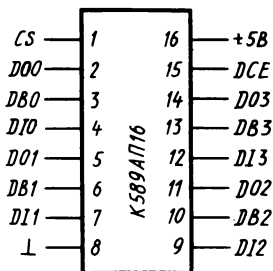


Рис. 2.6. Цоколевка микросхемы К589АП16

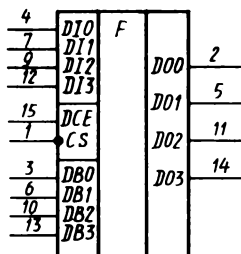


Рис. 2.7. Условное обозначение микросхемы К589АП16

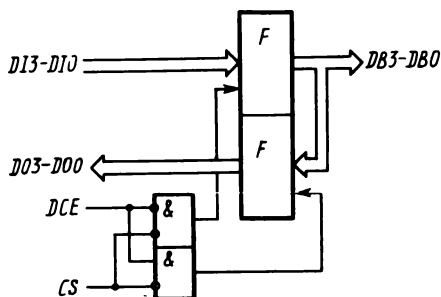


Рис. 2.8. Функциональная схема К589АП16

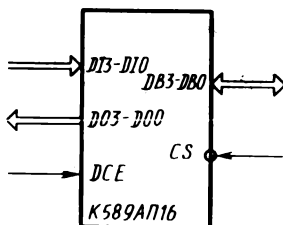


Рис. 2.9. Схема подключения К589АП16

вход управления заземлен, а вход CS подключен ко входу РМ, что позволяет отключать адресную магистраль в режиме ПДП, так же как и магистраль данных.

Микросхемы КР580ВА86 и КР580ВА87 представляют собой 8-разрядные двунаправленные формирователи, обеспечивающие токи нагрузки 32 мА. Они имеют двунаправленный канал А, подключаемый к микропроцессору, и двунаправленный канал В, подключаемый к системной магистрали (рис. 2.12—2.15). Формирователи имеют выходы с тремя состояниями и управляются сигналами направления передачи данных и отключения, подаваемыми на входы TF и OE соответственно. Передача информации разрешена при $\overline{OE} = 0$; причем если $TF = 1$, то направление передачи $A \rightarrow B$, если $TF = 0$; то передача имеет направление $B \rightarrow A$. При $\overline{OE} = 1$ выходы каналов А и В переводятся в третье состояние.

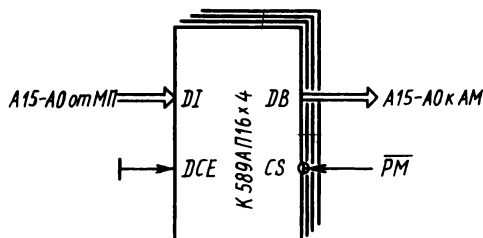


Рис. 2.10. Схема включения K589AP16 в качестве формирователя адресной магистрали

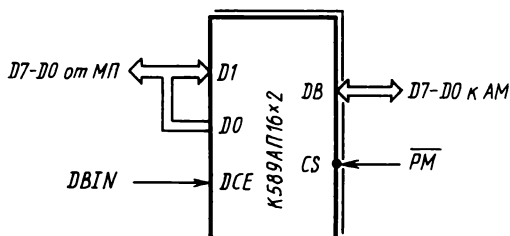


Рис. 2.11. Схема включения K589AP16 в качестве формирователя магистрали данных

В адресном буфере (рис. 2.16) вход TF подключен к цепи логической 1. В формирователе магистральных данных (рис. 2.17) на вход TF подается инвертированный сигнал с выхода DBIN микропроцессора КР580ВМ80А.

Сигналы управления формируются на основе байта состояния, выдаваемого микропроцессором по магистрали данных и сигналов управления DBIN и \overline{WR} . Для запоминания и хранения байта состояния используется регистр состояния, в качестве которого применяются различные интегральные схемы регистров или триггеров. Наиболее универсальными являются БИС 8-разрядных регистров K589ИР12 и КР580ИР82 (КР580ИР83).

Микросхема многорежимного буферного регистра K589ИР12, представленная на рис. 2.18 и 2.19, содержит 8-разрядный регистр; триггер прерываний; логические вентили управления режимами работы (рис. 2.20).

Регистр имеет два режима работы, определяемые состоянием входа MD. Если MD = 0, то осуществляется режим ввода, если MD = 1, то выполняется режим вывода. В режиме ввода информация с входного канала

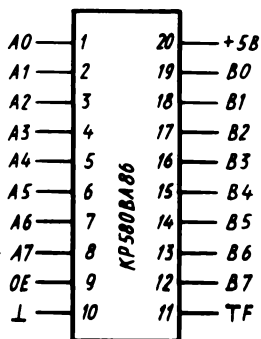


Рис. 2.12. Цоколевка микросхемы KP580BA86

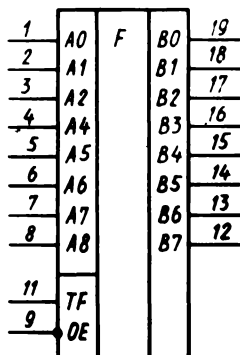


Рис. 2.13. Условное обозначение микросхемы KP580BA86

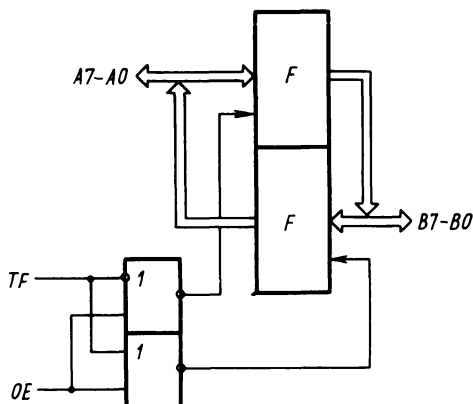


Рис. 2.14. Функциональная схема KP580BA86

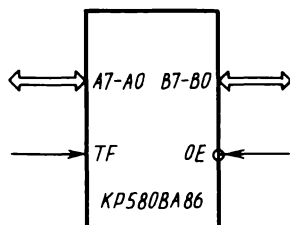


Рис. 2.15. Схема подключения KP580BA86

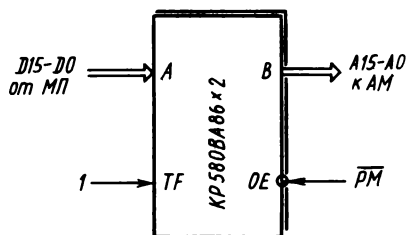


Рис. 2.16. Схема включения КР580ВА86 в качестве формирователя адресной магистрали

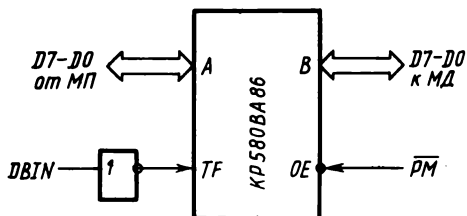


Рис. 2.17. Схема включения КР580ВА86 в качестве формирователя магистрали данных

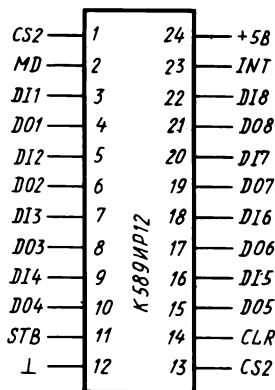


Рис. 2.18. Цоколевка микросхемы К589ИР12

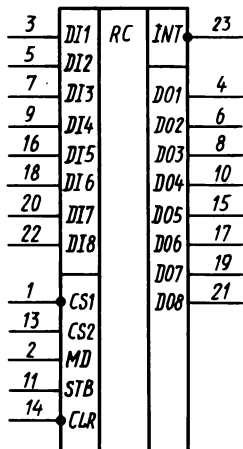


Рис. 2.19. Условное обозначение микросхемы К589ИР12

DI7 — DI0 заносится в регистр по стробу на входе STB. Информация выдается по каналу DO7 — DO0 по сигналам на входах CS1 и CS2 (рис. 2.21).

В режиме вывода выходные формирователи открыты и информация выдается в канал DO7 — DO0 постоянно. Занесение информации в регистр происходит по сигналам

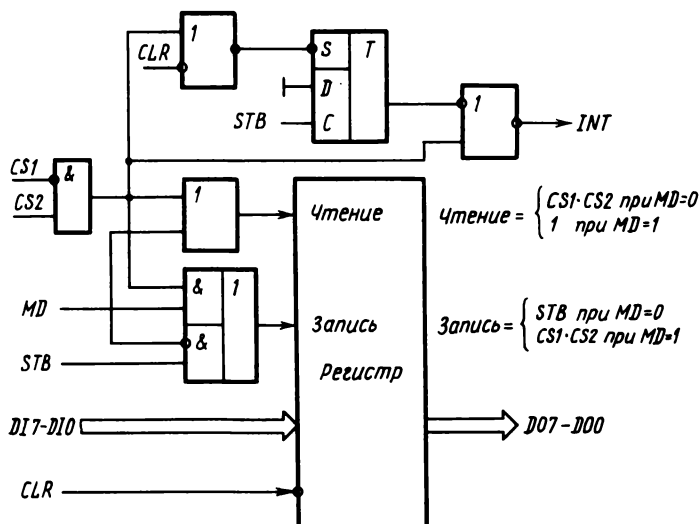


Рис. 2.20. Функциональная схема К589ИР12

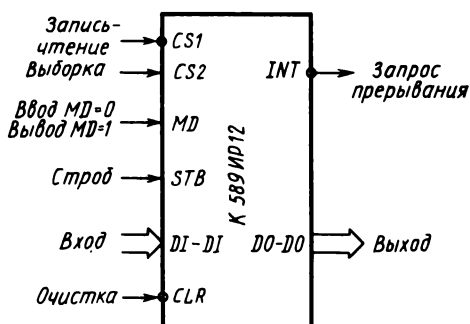


Рис. 2.21. Схема подключения К589ИР12

на входах $\overline{CS1}$ и $CS2$. Регистр может устанавливаться в нулевое состояние при подаче сигнала 0 на вход \overline{CLR} .

Триггер прерывания принимает и хранит сигналы, которые используются в качестве запросов прерывания. Сигналы поступают на вход синхронизации триггера со входа STB . Запрос прерывания выдается через выход \overline{INT} . В исходном состоянии, установленном сигналом на входе \overline{CLR} , на выходе \overline{INT} формируется 1, указывающая

на отсутствие запроса прерывания. Под действием сигнала на входе \overline{STB} триггер устанавливается в 0, а на выходе \overline{INT} появляется сигнал запроса прерывания нулевого уровня. Сброс запроса прерывания осуществляется сигналами выборки $\overline{CS1}$ и $\overline{CS2}$. При подаче сигналов $\overline{CS1}$ и $\overline{CS2}$ триггер прерывания устанавливается в 1, однако сигнал запроса прерывания на выходе \overline{INT} сохраняется до окончания сигналов выборки.

В качестве регистра состояния микросхема К589ИР12 работает в режиме вывода ($MD = 0$), поэтому байт состояния заносится непосредственно с магистрали данных по сигналам $F1_{TTL}$ и \overline{SYNC} , подаваемым на входы $\overline{CS1}$ и $\overline{CS2}$.

Микросхема КР580ИР82 представляет собой 8-разрядный регистр с управляемыми выходными формирователями (рис. 2.22—2.24). Запись информации происходит по стробу, подаваемому на вход \overline{STB} ; чтение информации по выходам \overline{DO} возможно при $OE = 0$. При использо-

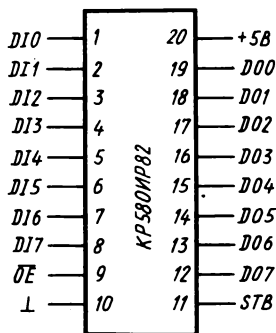


Рис. 2.22. Цоколевка микросхемы КР580ИР82

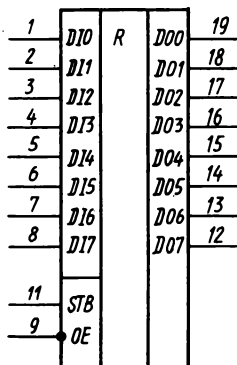


Рис. 2.23. Условное обозначение микросхемы КР580ИР82

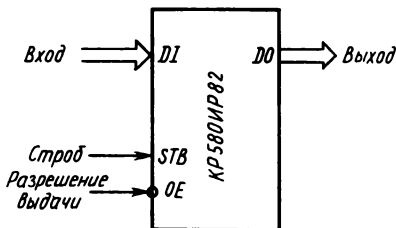


Рис. 2.24. Схема подключения КР580ИР82

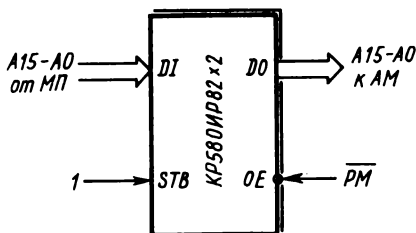


Рис. 2.25. Схема включения КР580ИР82 в качестве адресного буфера

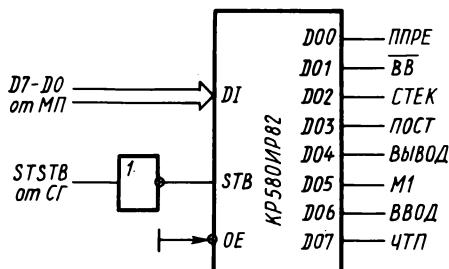


Рис. 2.26. Схема включения КР580ИР82 в качестве регистра состояния

вании КР580ИР82 в качестве регистра состояния (рис. 2.25, 2.26) необходимо установить $OE=0$, а на вход \overline{STB} подать строб состояния, формируемый системным генератором КР580ГФ24 или специальной логической схемой.

Так как адреса ячеек памяти и устройств ввода — вывода передаются по общей адресной магистрали, они разделяются сигналами управления $\overline{ЧТП}$, $\overline{ЧТВВ}$, $\overline{ЗПП}$, $\overline{ЗПВВ}$, формируемыми на основе битов состояния $\overline{ЧТ}$, $\overline{ВЫВ}$, $\overline{ВВОД}$ и сигналов с выходов \overline{DBIN} и \overline{WR} микропроцессора. Реализация схемы формирования управляющих сигналов показана на рис. 2.27.

Сигнал подтверждения прерывания предназначен для разрешения ввода команды модуля RST с магистрали данных, поступающих от модуля ввода — вывода. Он формируется по сигналу с выхода \overline{DBIN} при наличии бита ППРЕ в байте состояния. Другие сигналы баята состояния используются для построения системы контроля и отладки микроЭВМ.

Для упрощения интерфейса центрального процессора разработана специальная БИС системного контроллера КР580ВК28 и КР580ВК38 (рис. 2.28), осуществляющая формирование выходных сигналов магистрали данных и сигналов управления ЗУ и ВУ. Микросхема содержит двунаправленный магистральный формирователь МФ,

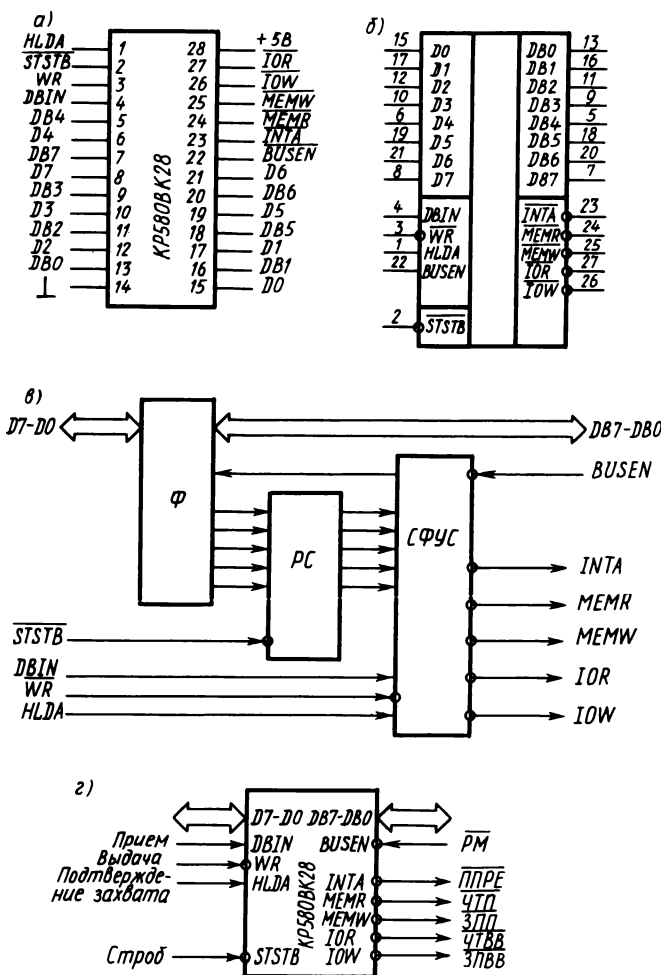


Рис. 2.28. Микросхема KP580BK28:

а — цоколевка; б — условное обозначение; в — функциональная схема;
г — назначение выводов

регистр состояния (PC) и схему формирования управляющих сигналов (СФУС). Формирователь управляется сигналами, подаваемыми на входы DBIN, HLDA с соответствующих выходов KP580ИК80А, и внешним сигналом разрешения магистралей, подаваемым на вход BUSEN с системы ПДП. Байт состояния запоминается

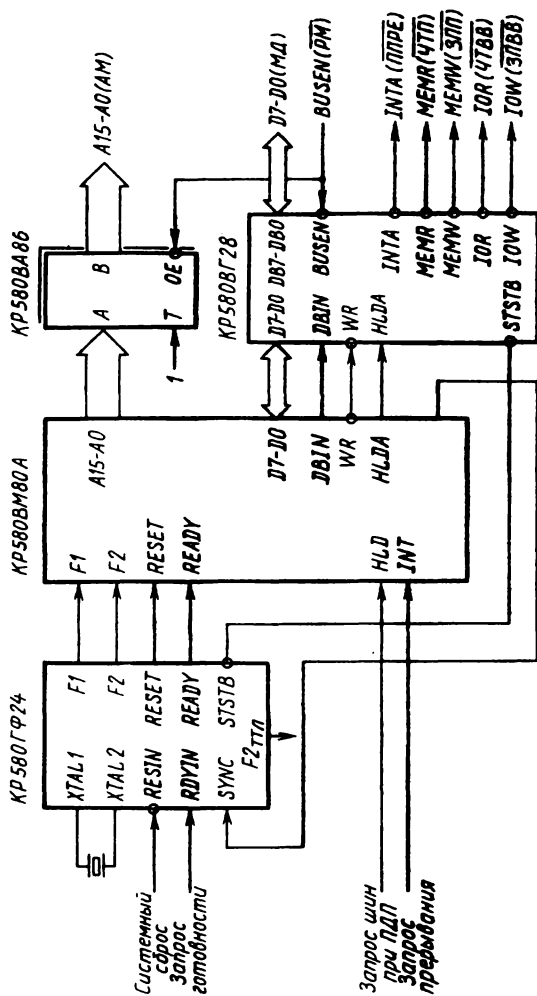


Рис. 2.29. Структура центрального процессора на БИС К580

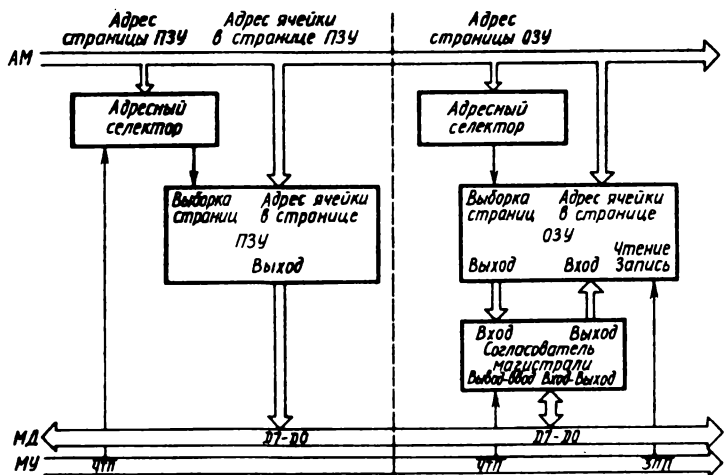


Рис. 2.30. Структура страниц внутренней памяти ЦП на БИС КР580ВМ80А

по стробу, подаваемому на вход \overline{STSTB} с одноименного выхода КР580ГФ24.

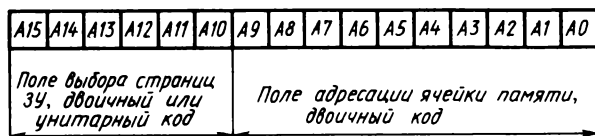
Микросхема формирует сигналы $\overline{ЧТП}$, $\overline{ЗПП}$, $\overline{ЧТВВ}$, $\overline{ЗПВВ}$ и $\overline{ППРЕ}$. Реализация центрального процессора на БИС семейства К580 показана на рис. 2.29.

Резидентные запоминающие устройства процессора — оперативное (ОЗУ) и постоянное (ПЗУ) — реализуются в области адресов общей памяти, занимая объем 2—8 К байт. Память микроЭВМ имеет страничную организацию. Резидентное ЗУ занимает одну или несколько страниц. Эта область должна быть недоступна прикладным программам, а адресоваться лишь со стороны системного программного обеспечения (ПО).

Страница ЗУ состоит из накопителя, осуществляющего хранение информации, и интерфейса ЗУ, реализующего функции сопряжения с системной магистралью (рис. 2.30). Накопитель строится обычно на одной или нескольких БИС ЗУ, обеспечивающих требуемый объем страницы и разрядность хранимых слов; БИС ПЗУ обеспечивает хранение многоразрядных слов (обычно байтов); их число в странице при байтовом доступе определяется объемом страницы. БИС ОЗУ, как правило, имеют организацию $N \text{ слов} \times 1 \text{ бит}$. Их число в странице определяется требуемой разрядностью и объемом страницы. Для упрощения интерфейса с системной маги-

стралью внутреннее ОЗУ центрального процессора реализуется обычно на БИС статического типа. Интерфейс ЗУ осуществляет выделение адресов страницы, управление чтением — записью.

Адресный формат при объеме страницы в 1 К байт, например, имеет вид



Поле адресации ячеек памяти кодируется внутри страницы двоичным кодом и дешифруется внутренними цепями БИС памяти. Страницы выбираются адресным селектором. При двоичном кодировании селектор реализуется на основе дешифраторов. При унитарном кодировании используется линейный выбор страниц, при котором каждый разряд адреса непосредственно выбирает страницу памяти. Это позволяет упростить интерфейс памяти, однако объем адресной памяти уменьшается.

Сопряжение БИС ЗУ с адресной магистралью осуществляется или непосредственным подключением адресных входов БИС памяти, или через магистральные формирователи. Сопряжение БИС памяти с системной магистралью данных наиболее просто реализуется для ПЗУ, имеющих выход с тремя состояниями или с открытым коллектором. Обычно выходы ПЗУ непосредственно подключаются к системной магистрали данных. В некоторых случаях для обеспечения требуемой нагрузочной способности используются магистральные формирователи. Аналогично согласуются ОЗУ с двунаправленной магистралью данных. Для сопряжения ОЗУ с отдельными входами и выходами данных используются согласователи, которые, в свою очередь, решают и задачи согласования нагрузок. Эти функции реализуются с помощью магистральных формирователей или буферных регистров.

Память управляется сигналами $\overline{\text{ЧТП}}$ для ПЗУ и $\overline{\text{ЧТП}}$, $\overline{\text{ЗПП}}$ для ОЗУ. Обычно ПЗУ не имеет отдельного входа управления чтением, поэтому линия $\overline{\text{ЧТП}}$ подключается к одному из входов выборки микросхем БИС ПЗУ или ко входу выборки адресного селектора. При использо-

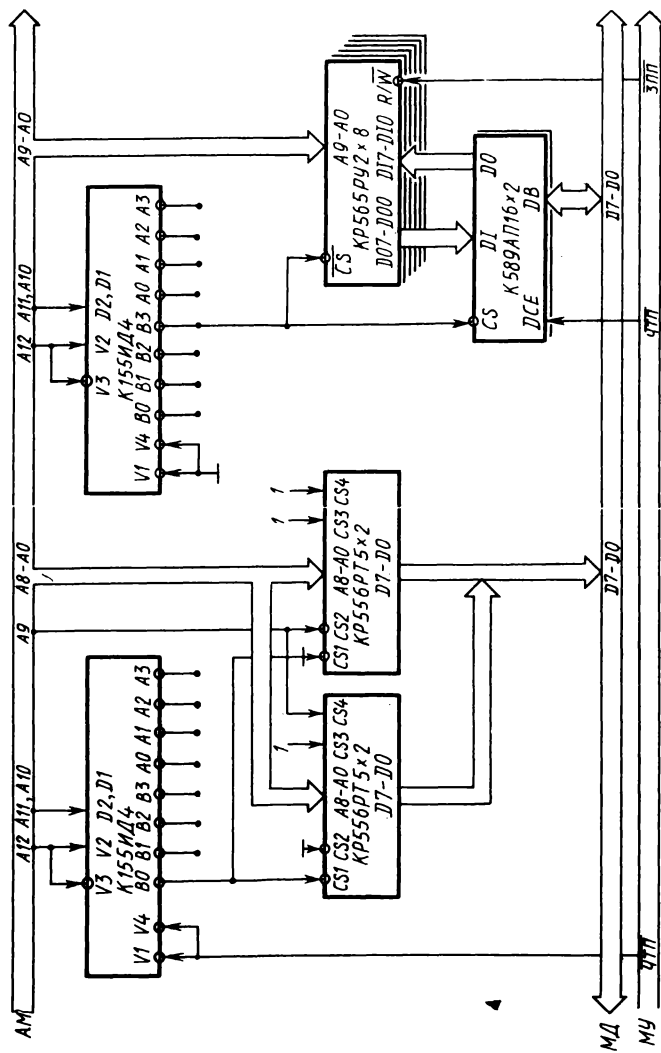


Рис. 2.31. Структура внутренней памяти ЦП на БИС KR580BM80A

вании формирователей линия ЧТП подключается ко входу управления магистрального формирователя. Как правило, ОЗУ имеет общий вход управления чтением — записью и на него подают сигнал управления записью. Сигнал управления чтением при этом используется для переключения на вывод согласователя магистралей ОЗУ с отдельными входными и выходными магистралями данных.

Пример реализации внутреннего ЗУ центрального процессора на БИС ОЗУ КР565РУ2 и БИС ППЗУ КР556РТ5 показан на рис. 2.31. В этом примере объем страницы составляет 1 К байт. Запоминающее устройство располагается в пространстве адресов в зависимости от положения переключателей. Страница ПЗУ реализована на двух БИС КР556РТ5, страница ОЗУ — на БИС КР565РУ2. Страницы выбираются адресными селекторами, реализованными на интегральных схемах дешифраторов К155ИД4. Каждая страница располагается в области адресов 0 000—0FFF.

Магистраль данных ОЗУ согласуется с системной магистралью данных с помощью двух интегральных схем К589АП16.

2.2. Проектирование центрального процессора на БИС семейства К1810

Два режима работы микропроцессора КМ1810ВМ86 позволяют строить на его основе центральные процессоры для простых однопроцессорных микроЭВМ и микроконтроллеров и вычислительные модули для мультипроцессорных систем [6].

При построении простых микроЭВМ и микроконтроллеров центральный процессор имеет в своем составе микропроцессор КМ1810ВМ86, БИС системного генератора КР1810ГФ84 и БИС магистрального интерфейса КР580ИР82 и КР580ИР83, КР580ВА86 и КР580ВА87 или КР1810ИР82 и КР1810ИР83, КР1810ВА86 и КР1810ВА87.

При построении вычислительных модулей мультипроцессорных систем требуются системный контроллер КР1810ВГ88 и арбитр магистралей КР1810ВБ89.

Система синхронизации реализуется на основе микросхемы КР1810ГФ84, предназначенной для формирования сигналов синхронизации микропроцессора КМ1810ВМ86,

сигналов синхронизации системной магистрали, внешних сигналов готовности и начальной установки (рис. 2.32, 2.33).

Сигналы синхронизации микропроцессора на выходе CLK формируются задающим генератором, стабилизируемым внешним кварцевым резонатором частотой 12—25 МГц, и делителем (Д) частоты на 3 (рис. 2.34). Для получения сигналов синхронизации на выходе PCLK используется делитель на 2. Можно использовать внешний задающий генератор, подключаемый по входу

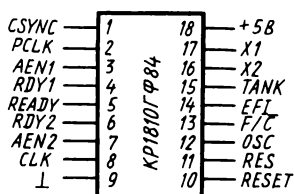


Рис. 2.32. Цоколевка микросхемы KP1810GF84

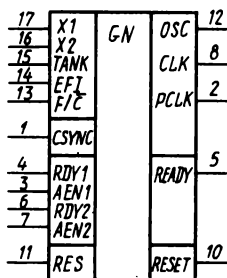


Рис. 2.33. Условное обозначение микросхемы KP1810GF84

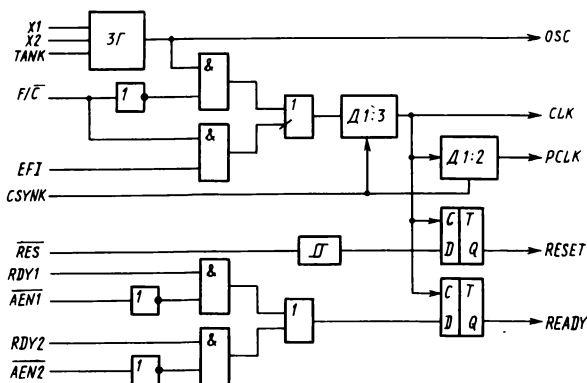


Рис. 2.34. Функциональная схема микросхемы KP1810GF84

EFI. Управление выборкой генератора осуществляется внешним сигналом по входу F/C. Задающий генератор имеет выход высокой частоты OSC и вход управления высшими гармониками кварцевого резонатора TANK.

Делители имеют входы внешней синхронизации CSYNK, что позволяет осуществлять синхронизацию с другими генераторами в системе. Сигнал высокого уровня на входе CSYNK сбрасывает счетчики, низкого — разрешает работу. При работе с внешним задающим генератором вход используется; при работе с внутренним задающим генератором вход заземляется (рис. 2.35—2.37). Временные диаграммы формирования синхросигналов показаны на рис. 2.38.

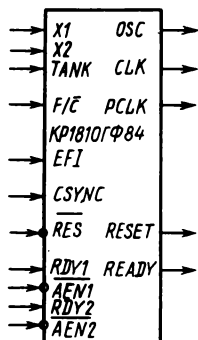


Рис. 2.35. Схема подключения KP1810GF84

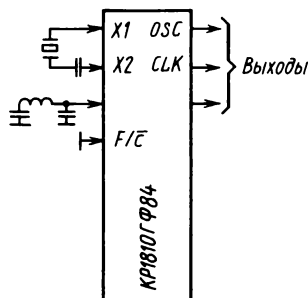


Рис. 2.36. Схема включения KP1810GF84 с внутренним задающим генератором

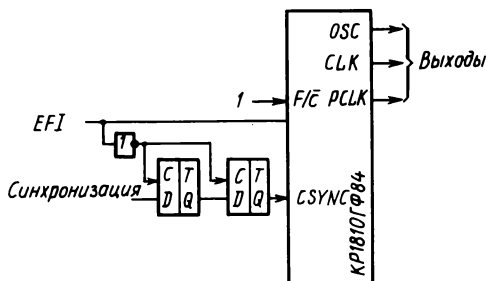


Рис. 2.37. Схема включения KP1810GF84 с внешним задающим генератором

Синхронизация внешних сигналов начальной установки микропроцессора осуществляется с помощью триггера Шмидта и запоминающего триггера, синхронизируемого сигналами CLK. Сигнал начальной установки на входе \overline{RES} синхронизируется спадом синхросигнала CLK (рис. 2.38).

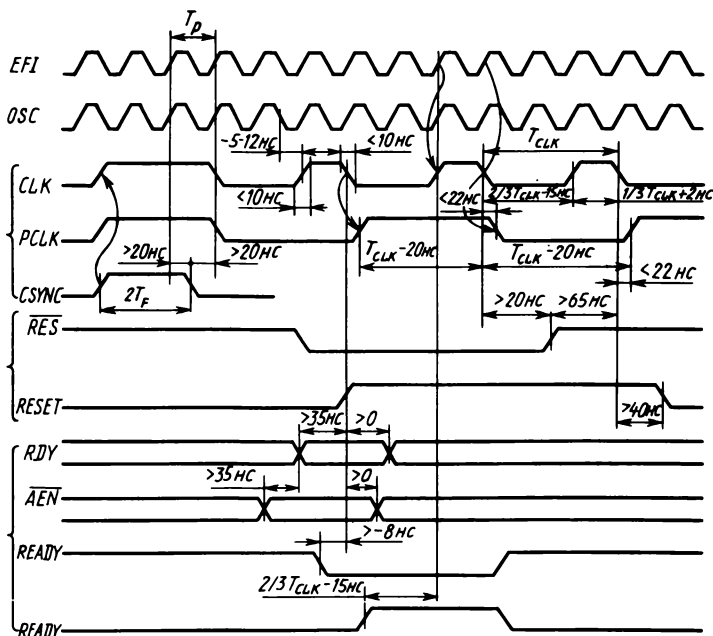


Рис. 2.38. Временные диаграммы работы микросхемы КР1810ГФ84

Микросхема имеет два входа готовности PDY1, PDY2, позволяющие принимать запросы ожидания от двух системных магистралей. Входы выбираются внешними сигналами, подаваемыми на входы адресов готовности $\overline{AEN1}$, $\overline{AEN2}$. Временные диаграммы формирования сигналов готовности показаны на рис. 2.38.

Интерфейс центрального процессора, выполненного на БИС КМ1810ВМ86, реализуется на БИС КР1810ИР82, КР1810ИР83, КР1810ВА86, КР1810ВА87.

Микросхемы КР1810ИР82, КР1810ИР83 используются в качестве буферных регистров адреса для демultipлексирования адресной магистрали микропроцессора; микросхемы КР1810ВА86, КР1810ВА87 — в качестве формирователей магистрали данных.

Обозначение микросхем и их структура приведены в § 2.1.

Контроллер магистрали реализуется на основе микросхемы КР1810ВГ88 (рис. 2.39—2.41), предназначенной для формирования сигналов управления обменом между

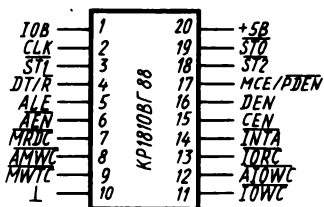


Рис. 2.39. Цоколевка микросхемы KR1810BG88

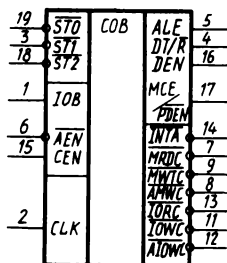


Рис. 2.40. Условное обозначение микросхемы KR1810BG88

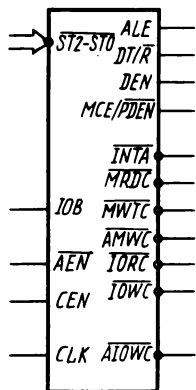


Рис. 2.41. Назначение выводов микросхемы KR1810BG88

локальной магистралью процессора, системной магистралью или магистралью ввода — вывода. Контроллер формирует следующие командные сигналы управления системной магистралью и магистралью ввода — вывода:

\overline{INTA} — подтверждение прерывания, используется для чтения вектора прерывания;

\overline{MRDC} — чтение памяти;

\overline{IORC} — чтение устройств ввода — вывода;

\overline{MWTC} — запись в память;

\overline{AMWC} — опережающая запись в память;

\overline{IOWC} — запись в устройство ввода — вывода;

\overline{AIOWC} — опережающая запись в устройства ввода — вывода.

Управление интерфейсом процессора осуществляется сигналами, формируемыми на выходах:

ALE — строб адреса, используется для загрузки адреса в буферный регистр адреса;

DT/R — направление передачи данных, при $DT/\overline{R} = 0$ — чтение или ввод данных; $DT/\overline{R} = 1$ — запись или вывод данных. Используется для коммутации формирователей магистрали данных на ввод или вывод;

DEN — разрешение данных, управляет состоянием формирователей магистрали данных; DEN=0 — формирователи отключены от магистрали данных; DEN=

= 1 — передача данных через формирователи разрешена;

$\overline{MCE}/\overline{PDEN}$ — разрешение ведущего контроллера прерываний при каскадировании ($IOB = 0$) или разрешение данных периферийных устройств ($IOB = 1$). Таким образом, выполняются две функции: при $IOB = 0$ формируется сигнал разрешения чтения вектора прерывания ведущего контроллера прерываний; при $IOB = 1$ выход используется для выдачи сигнала разрешения данных, поступающих от внешних устройств через резидентную системную магистраль процессорного модуля.

Выходные сигналы контроллера формируются на основе кода состояния, подаваемого на выходы $\overline{ST2}$ — $\overline{ST0}$ с соответствующих выходов микропроцессора. Сигналы синхронизируются общими с микропроцессором синхроимпульсами, подаваемыми на вход CLK (рис. 2.42).

Выходные сигналы контроллера зависят также от сигналов управления, подаваемых на входы:

IOB — режим работы;

\overline{AEN} — адресное разрешение;

\overline{CEN} — разрешение команд.

Микросхема может работать в двух режимах, задаваемых сигналами на входе IOB .

При $IOB = 0$ микросхема работает в режиме контроллера системной магистрали. Все выходы сигналов управления работой системной магистрали управляются сигналом адресного разрешения на входе \overline{AEN} : при $\overline{AEN} = 1$ выходы сигналов находятся в третьем состоянии; с появлением сигнала на входе $\overline{AEN} = 0$ через 115 нс вырабатываются сигналы на входах управления работой системной магистрали (рис. 2.43), что позволяет управлять контроллером при работе с многопользовательской системной магистралью.

Сигнал на входе \overline{AEN} формируется арбитром магистрали. Контроллер обслуживает одну системную магистраль. Если вычислительный модуль предназначен для работы с двумя магистралями: многопользовательской общесистемной и резидентной, то для управления каждой из них используется свой контроллер. Разделение адресных пространств резидентной и общесистемной памяти при этом осуществляется сигналом разрешения команд на входе \overline{CEN} (рис. 2.43). При $\overline{CEN} = 1$ все выходы команд контроллера находятся в активном состоянии, контроллер формирует команды управления системной

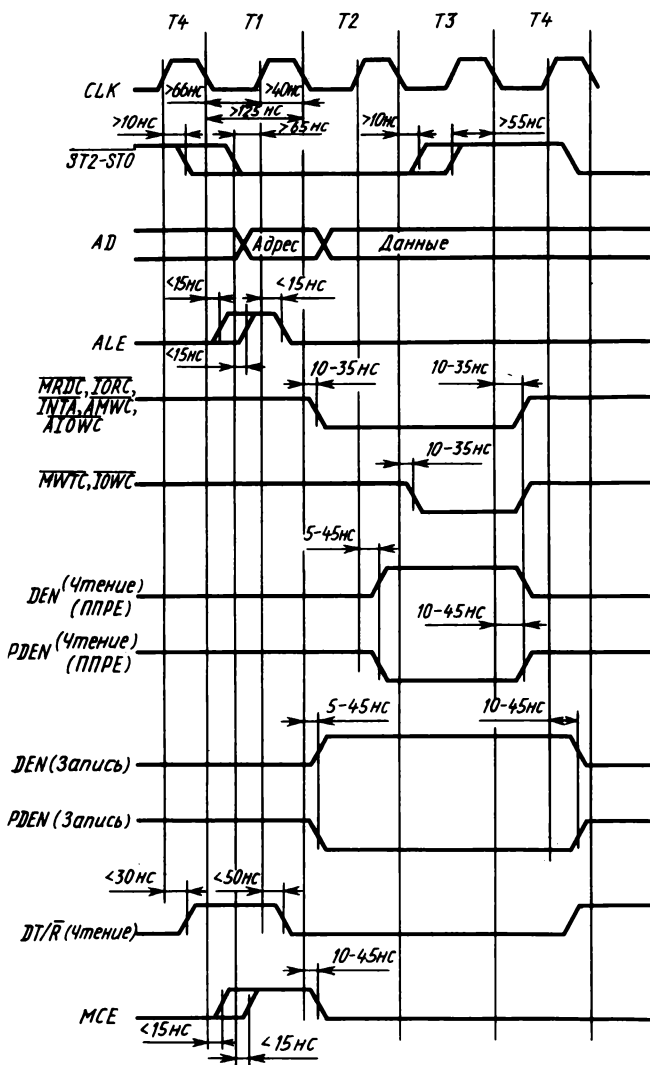
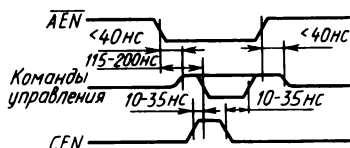


Рис. 2.42. Временные диаграммы работы микросхемы КР1810ВГ88

Рис. 2.43. Временные диаграммы управления сигналами команд микросхемы КР1810ВГ88



магистралью. При $CEN = 0$ все выходы команд находятся в неактивном состоянии, контроллер теряет управляющие функции. Обычно сигнал на входе CEN формируется адресным дешифратором, расположенным на резидентной магистрали. Для адресов резидентной памяти дешифратор формирует сигнал разрешения контроллера резидентной магистрали ($CEN = 1$) и сигнал запрета контроллера системной магистрали $CEN = 0$. Для адресов общесистемной памяти сигналы управления на входах CEN контроллеров имеют инверсные значения. Приоритетное управление контроллером общесистемной магистрали, как и ранее, осуществляется арбитром по входу \overline{AEN} .

В режиме контроллера магистрали ввода — вывода ($IOB = 1$) выходы сигналов управления UBB не зависят от состояния сигнала на входе \overline{AEN} — они всегда разрешены, а разрешение формирователя магистрали данных устройств ввода — вывода осуществляется сигналом на выходе $PDEN$. Линии команд управления вводом — выводом не могут использоваться для работы с многопользовательской системной магистралью ввода — вывода. Выходы команд управления памятью по-прежнему зависят от состояния сигнала на входе \overline{AEN} . Контроллер может обслуживать две магистрали: общесистемную многопользовательскую и ввода — вывода.

На общесистемной магистрали располагается общая память, на магистрали ввода — вывода — внешние устройства.

Функциональная схема центрального процессора для одноплатных микроЭВМ и микроконтроллеров показана на рис. 2.44. Микропроцессор работает в минимальном режиме ($MN/\overline{MX} = 1$) и обеспечивает доступ к памяти объемом 1 М байт с организацией $512K \times 16$. Интерфейс процессора реализуется на микросхемах регистра КР1810ИР82, КР1810ИР83 и магистральных формирователях КР1810ВА86, КР1810ВА87. Для формирования

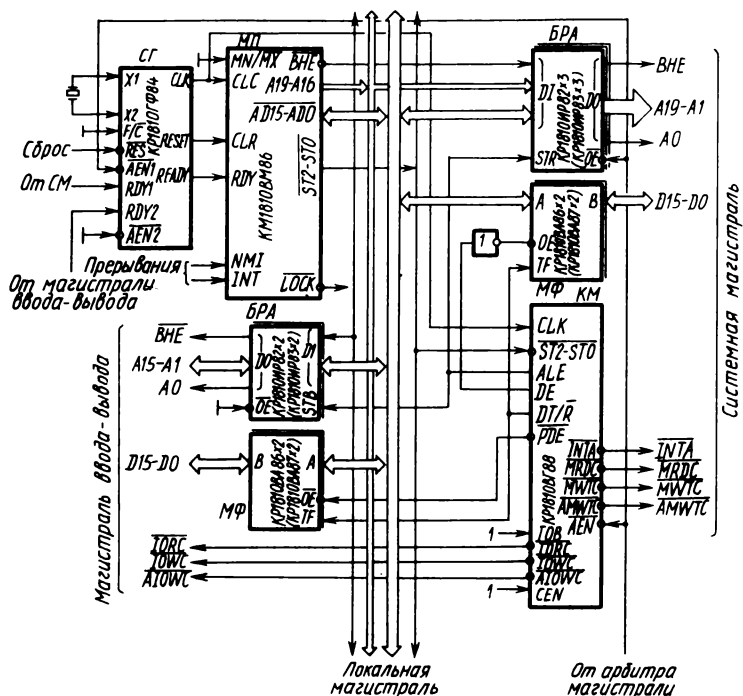


Рис. 2.46. Структура вычислительного модуля на БИС КМ1810ВМ86 с индивидуальной магистралью ввода — вывода

доступом к магистральной осуществляется сигналом, подаваемым на вход «Адресное разрешение». Этот сигнал формирует арбитр магистральной. К системной магистральной подключаются общесистемные ресурсы — память и внешние устройства, а также другие вычислительные модули.

Вычислительный модуль, приведенный на рис. 2.46, имеет автономную магистраль ввода — вывода, управляемую общим контроллером магистральной КР1810ВГ88. Контроллер работает в режиме управления вводом — выводом (IOB=1). Интерфейс магистральной ввода — вывода также реализован на интегральных схемах КР1810ИР82, КР1810ИР83, КР1810ВА86, КР1810ВА87. Магистраль однопользовательская, поэтому вход разрешения адресного буфера заземлен, а буфер магистральной данных управляется сигналом с выхода PDEN контроллера.

память и внешние устройства модуля. Общесистемная магистраль — многопользовательская, к ней кроме общесистемной памяти и устройств ввода — вывода могут подключаться другие вычислительные модули. Управление каждой магистралью осуществляется отдельным контроллером КР1810ВГ88.

Адресное пространство резидентной магистрали выделяется адресным дешифратором, который формирует сигнал, управляющий доступом к устройствам на системной или резидентной магистрали.

глава 3

Представители микроЭВМ «Электроника К1»



Наиболее характерными микроЭВМ, использующими в полном объеме возможности рассмотренных микропроцессоров, являются представители семейства «Электроника К1». В главе приводятся основные структурные решения, конструктивные особенности, излагается функционирование центрального процессора и других модулей, входящих в микроЭВМ.

3.1. МикроЭВМ «Электроника К1-30».

Назначение, состав и структура

Эффективное использование микроЭВМ в значительной степени определяется сроками и качеством разработки программного обеспечения (ПО). Один из наиболее рациональных способов проектирования ПО — применение отладочных комплексов [4, 5]. Отладочный комплекс, как правило, представляет собой конструктивно законченный блок, объединяющий в своем составе 2—15 плат и имеющий пульт. Основу комплекса составляют микроЭВМ (на базе комплекта БИС, применяемого в проектируемой системе) и системная магистраль, которая выводится на внешний разъем для подключения контроллеров, разрабатываемых пользователем. МикроЭВМ, являющаяся основой отладочного комплекса, может применяться самостоятельно. Комплексы имеют интерфейсы для подключения внешних устройств. Их состав зависит от варианта исполнения и определяется пользователем, который закупает и подключает необходимые в каждом конкретном случае устройства. Для разработки ПО пользователю предоставляется система программирования, поддерживаемая операционной системой комплекса.

Семейство «Электроника К1» состоит из комплекса технических средств (КТС) «Электроника К1-10», «Электроника К1-30» («Электроника МС 0401») и кон-

троллера программируемого универсального «Электроника К1-20» («Электроника МС 2702»).

Комплекс технических средств — отладочный комплекс, предназначен для использования в качестве рабочих мест отладки программных средств пользователя, разрабатываемых для аппаратуры на базе БИС серий К580, К1810, К1816.

МикроЭВМ, входящая в состав КТС, может применяться в системах контроля и управления технологическими процессами и промышленными установками, в информационно-измерительных системах и системах автоматизации научного эксперимента.

Исполнения комплекса в соответствии с основным своим назначением отличаются различными составами функциональных модулей, емкостями внутренних запоминающих устройств, составом программ математического обеспечения, а также комплектом поставляемых периферийных приборов.

Некоторые варианты исполнения КТС «Электроника К1-30» приведены на рис. 3.1. Базовый комплект имеет в своем составе следующие типовые модули: центральный процессор ЦП; устройство сопряжения УС с пультом управления УС ПУ; пульт управления ПУ; устройство сопряжения с фотоэлектрическим устройством ввода с

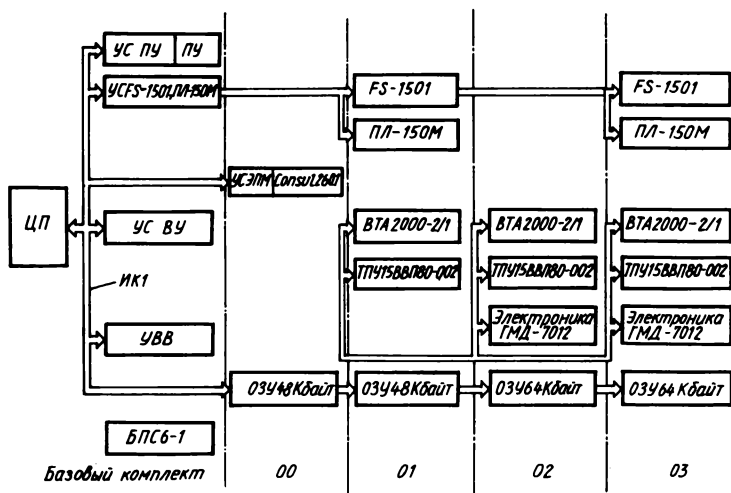


Рис. 3.1. Варианты исполнения комплекса технических средств

перфоленты и перфоратором ленточным УС FS-1501, ПЛ-150М; устройство сопряжения с внешними устройствами УС ВУ; устройство ввода — вывода УВВ и блок питания БПС6-1.

Исполнения КТС (от 00 до 03), приведенные на рисунке, отличаются: 1) комплектностью периферийных приборов, таких, как FS-1501, ПЛ-150М, видеотерминал алфавитно-цифровой ВТА2000-2/1 или 15ИЭ200×140-017, устройство ввода — вывода на ГМД «Электроника ГМД-7012», термопечатающее устройство ТПУ15ВВП80-002 или мозаичная печать УВВПЧ-30-004; 2) типом и емкостью ОЗУ, в качестве которого может быть использовано ОЗУ объемом 48 К байт или ОЗУ объемом 64 К байт; 3) наличием или отсутствием электрофицированной пишущей машинки ЭПМ типа «Consul 260.1» и соответствующего устройства сопряжения УС ЭПМ.

Обмен данными между устройствами, входящими в состав данной микроЭВМ, обеспечивается интерфейсом ИК1. Структурные особенности, интерфейс и система команд устройства ЦП и микроЭВМ определяются микропроцессором КР580ВМ80А, используемым в центральном процессоре.

Комплекс технических средств «Электроника К1-30» («Электроника МС 0401») имеет следующие основные технические характеристики:

Элементная база — микросхемы серий К580, К589, К565, К155, К505. Базовый комплект — БИС серии К580. Центральный процессор — КР580ВМ80А

Разрядность адреса — 16 бит

Разрядность данных — 8, 16 бит

Число команд — 78 (111 операций)

Форматы команд — нуль-, одно- и двухадресные

Разрядность команд — 8, 16, 24 бит

Формат данных — двоичные числа с фиксированной запятой

Способы адресации — прямая, косвенная, непосредственная, неявная

Число 8-разрядных регистров общего назначения — 6

Время выполнения команд сложения R-R — не более 2 мкс

Число адресуемых регистров ввода — вывода — 256

Объем адресуемой памяти — 64 К байт

Емкость ПЗУ — 16 К байт; ОЗУ — 64 К байт

Емкость внешней памяти — до 1 М байт (накопители на гибких дисках)

Производительность — 110 тыс. оп/с

Система прерываний — программноприоритетная с 8 линиями аппаратного прерывания

Время реакции на прерывание — 18 мкс

Число программируемых каналов ввода — вывода по типу интерфейса ИРПР — 72, скорость обмена — 50 К бит/с

Число программируемых последовательных каналов по типу ИРПС: ввода — 1, вывода — 1, скорость обмена — 56 000 бит/с

Число программируемых 16-разрядных таймеров: для пользователя — 1; для отладки программ — 2

Параметры входных и выходных сигналов совместимы с ТТЛ-уровнями:

Напряжение питания — 220 В (50 Гц)

Потребляемая мощность — не более 300 В·А

Габаритные размеры: 550×483×133 мм

Масса — 21 кг

В состав программного обеспечения входят: системные программные средства контроля; операционные системы МОС-2, ДОС с функциями управления вводом — выводом и выполнением программ пользователя; языки программирования: ассемблеры, языки высокого уровня, а также средства программирования: Редактор-П, Редактор-Д, Редактор Э, Ассемблер П, Ассемблер Д, БЭЙ-СИК, Ассемблер АСМ-86, Конвертор АСМ80—АСМ86, Ассемблер АСМ-48.

3.2. Устройство и работа составных частей микроЭВМ «Электроника К1-30»

Представители семейства «Электроника К1» имеют общие характерные особенности построения составных частей, входящих в микроЭВМ. Наиболее важными для понимания работы являются: интерфейс, центральный процессор, запоминающие устройства и устройства сопряжения.

Особенности интерфейса ИК1. Интерфейс ИК1 (рис. 3.1) микроЭВМ «Электроника К1-30» и «Электроника К1-10» — унифицированная система связей и сигналов между ЦП, запоминающими устройствами и устройствами управления периферийным оборудованием. Для обеспечения технической совместимости периферийные устройства или адаптеры связи, разрабатываемые для подключения к данной микроЭВМ, должны удовлетворять требованиям интерфейса.

Интерфейс ИК1 выполнен в виде магистрали передачи информации, функционально разделенной на две части: системную и вспомогательную. Использование системной магистрали определяется обязательными требованиями по организации связи. Вспомогательная магистраль не определяет основных функциональных характеристик

интерфейса и является не обязательной для применения.

Состав системной и вспомогательной магистралей приведен в табл. 3.1.

В любой операции интерфейса ИК1 участвуют два устройства, между которыми существуют отношения задатчика и исполнителя.

Магистраль адреса $\overline{MA0} - \overline{MA15}$ предназначена для

Таблица 3.1

Обозначение магистралей	Наименование	Число линий	ТТЛ-уровни (0 или 1)
Системная магистраль			
$\overline{MA0} - \overline{MA15}$	Магистраль адреса	16	0
$\overline{MD0} - \overline{MD15}$	Магистраль данных	8	0
Ком. Чт.ЗУ	Команда чтения ЗУ	1	0
Ком. ЗП.ЗУ	Команда записи ЗУ	1	0
Ком. Чт. В/В	Команда чтения УВВ	1	0
Ком. ЗП.В/В	Команда записи УВВ	1	0
ВЫДАЧА	Выдача	1	0
ГОТОВНОСТЬ	Готовность	1	0
ВЕКТ. ПРЕР. 1—8	Векторы прерывания	8	0
τ_2 (ТТЛ)	Синхросигналы τ_2 (ТТЛ)	2	1
Вспомогательная магистраль			
БЛОКИР. ОЗУ	Блокировка ОЗУ	1	1
ПРИЕМ	Прием	1	0
СИНХРО	Синхронизация	1	0
ЗАХВАТ	Захват	1	0
ПОДТВ. ЗАХВ.	Подтверждение захвата	1	0
ОЖИД.	Ожидание	1	0
ПОДТВ. ЗАПР.	Подтверждение запроса	1	1
ПРЕР.	прерывания		
ЗАП/ВЫВОД	Запись — вывод	1	1
СТЕК	Стек	1	1
ПОДТВ. ОСТ.	Подтверждение останова	1	1
ВЫВОД	Вывод	1	1
М1	Признак М1	1	1
ВВОД	Ввод	1	1
ЧТЕНИЕ	Чтение	1	1
ВЫХ. ТАЙМ. 2	Выход таймера 2	1	1
ВХ. ТАЙМ. 2	Вход таймера 2	1	1
РАЗР. ПРЕР.	Разрешение на прерывание	1	0

передачи адреса ячейки памяти, адреса периферийного устройства или его регистров и используется для выбора исполнителя.

Магистраль данных $\overline{MD0} - \overline{MD7}$ используется для передачи данных между задатчиком и исполнителем.

Линия команды чтения $\overline{ЗУ}$ предназначена для выдачи задатчиком сигнала $\overline{Ком. \text{ЧТ. ЗУ}}$. Наличие этого сигнала означает, что на магистрали адреса $\overline{MA0} - \overline{MA15}$ находится адрес ячейки $\overline{ЗУ}$, из которой производится чтение информации.

Линия команды записи $\overline{ЗУ}$ предназначена для выдачи задатчиком сигнала $\overline{Ком. \text{ЗП. ЗУ}}$. Наличие сигнала означает, что на магистрали адреса $\overline{MA0} - \overline{MA15}$ находится адрес ячейки $\overline{ЗУ}$, в которую производится запись.

Линия команды чтения $\overline{УВВ}$ предназначена для выдачи задатчиком сигнала $\overline{Ком. \text{ЧТ. В/В}}$. Наличие его означает, что на магистралях адреса $\overline{MA0} - \overline{MA7}$ и $\overline{MA8} - \overline{MA15}$ установлены два одинаковых кода адреса периферийного устройства или его регистра, из которого осуществляется ввод информации.

Линия команды записи устройства ввода — вывода предназначена для выдачи задатчиком сигнала $\overline{Ком. \text{ЗП. В/В}}$. Наличие его означает, что на магистрали адреса $\overline{MA0} - \overline{MA7}$ и $\overline{MA8} - \overline{MA15}$ установлены два одинаковых кода адреса устройства вывода, в которое производится вывод информации.

Линия выдачи предназначена для выдачи задатчиком сигнала $\overline{ВЫДАЧА}$. Наличие этого сигнала означает, что на магистрали данных $\overline{MD0} - \overline{MD7}$ находится предназначенная для записи информация. При выполнении операций записи в запоминающее или периферийное устройство сигнал $\overline{ВЫДАЧА}$ может использоваться для стробирования приема информации с магистрали данных в запоминающее или периферийное устройство.

Линия сброса предназначена для установки в исходное состояние с помощью сигнала $\overline{СБРОС}$ регистров и схем управления устройств, подключенных к интерфейсу ИК1. Этот сигнал формируется при включении питания или нажатии кнопки $\overline{СБРОС}$ на ПУ.

Линия готовности предназначена для перевода ЦП в режим ожидания в течение определенного времени после получения исполнителем одного из сигналов

Ком. ЧТ. ЗУ, Ком. ЗП. ЗУ, Ком. ЧТ. В/В, Ком. ЗП. В/В, если требуется согласование быстрогодействия исполнителя и задатчика при выполнении операции чтения или записи из (в) ЗУ или УВВ. Перевод ЦП в режим ожидания происходит по сигналу ГОТОВНОСТЬ, который выдается исполнителем. Длительность сигналов Ком. ЧТ. ЗУ, Ком. ЗП. ЗУ, Ком. ЧТ. В/В, Ком. ЗП. В/В увеличивается на время действия сигнала ГОТОВНОСТЬ.

Линии векторов прерываний предназначены для прерывания выполнения программы ЦП и вызова соответствующей подпрограммы обработки прерываний с помощью сигналов ВЕКТ. ПРЕР.1 — ВЕКТ. ПРЕР. 8, которые вырабатываются периферийными устройствами. Сигнал ВЕКТ. ПРЕР.1 имеет самый высокий приоритет, ВЕКТ. ПРЕР. 8 — самый низкий.

Синхросигналы t_2 (ТТЛ) предназначены для синхронизации схем различных устройств, входящих в состав микроЭВМ. Сигнал t_2 (ТТЛ) вырабатывается генератором фаз ЦП синхронно с фазой Ф2 микропроцессора (см. гл. 1).

Сигналы вспомогательной магистрали интерфейса ПРИЕМ, СИНХРО, ПОДТВ.ЗАХВ., ОЖИД., РАЗР. ПРЕР. соответствуют сигналам микропроцессора КР580ВМ80А, а сигналы ПОДТВ. ЗАПР. ПРЕР., ЗАП./ВЫВОД, СТЕК, ПОДТВ. ОСТ., ВЫВОД, М1, ВВОД, ЧТЕНИЕ — выходным сигналам регистра состояний микропроцессора (см. § 2.1).

Сигнал блокировки ОЗУ используется для блокировки обращения к единому полю адресов ПЗУ и ОЗУ.

Сигнал приема используется для стробирования выдачи данных из ЗУ или УВВ на магистраль данных МД0 — МД7 с помощью сигнала ПРИЕМ, который выдается ЦП.

Линия захвата предназначена для выдачи сигнала ЗАХВАТ (запрос прямого доступа) пультом управления или периферийным устройством, подключенным к данной линии.

Интерфейс ИК1 предусматривает возможность одновременной выдачи только одного сигнала ЗАХВАТ устройством, работающим в режиме захвата, при этом данное устройство получает возможность управления интерфейсом и обращения к ЗУ или другому периферийному устройству. Центральный процессор может продолжить

выполнение программы только после снятия сигнала ЗАХВАТ.

Сигнал подтверждения захвата является разрешением захвата управления интерфейсом для устройства, вырабатывающего сигнал ЗАХВАТ. Сигнал ПОДТВ. ЗАХВ. выдается ЦП в ответ на сигнал ЗАХВАТ.

Линия ожидания предназначена для формирования сообщения центральным процессором о его переходе в состояние ожидания. При этом ЦП формирует сигнал ОЖИД. в ответ на сигнал ГОТОВНОСТЬ. Сигнал ОЖИД. может использоваться в схемах управления устройств, подключенных к интерфейсу ИК1.

На линии разрешения прерывания ЦП формирует сигнал РАЗР. ПРЕР., который означает, что процессор приступит к обслуживанию запросов на прерывание (при наличии какого-либо из сигналов ВЕКТ. ПРЕР.1 — ВЕКТ. ПРЕР. 8) после завершения выполнения текущей операции. При отсутствии сигнала РАЗР. ПРЕР. ЦП не реагирует на запросы на прерывание.

Сигналы микропроцессора ПОДТВ. ЗАПР. ПРЕР., ЗАП./ВЫВОД, СТЕК, ПОДТВ. ОСТ., ВЫВОД, М1, ВВОД, ЧТЕНИЕ предназначены для управления режимами работы и формирования сообщений о состояниях.

Линии ВХ. ТАЙМ.2 и ВЫХ. ТАЙМ.2 предназначены для управления работой таймера и использования программно-управляемых временных задержек в системе.

Обмен информацией между устройствами, подключенными к интерфейсу ИК1, по виду выполняемых действий распределяется на следующие основные группы: передача данных; прерывание программы; захват магистрали.

Существует четыре типа операций передачи данных по интерфейсу ИК1: чтение данных из памяти; запись данных в память; ввод данных из УВВ и вывод данных в УВВ.

Чтение данных из памяти осуществляется в соответствии с временными диаграммами, показанными на рис. 3.2, 3.3. Задатчик, управляющий интерфейсом ИК1, через интервал времени t_1 (рис. 3.2) после положительного фронта (переход $0 \rightarrow 1$) импульса t_2 (цикл T_1) формирует на магистрали МА0 — МА15 требуемый адрес и через промежуток времени t_2 вырабатывает сигнал Ком. ЧТ. ЗУ. По отрицательному фронту сигнала Ком. ЧТ. ЗУ записывающее устройство по адресу, установленному на магистраль МА0 — МА15, к моменту

появления положительного фронта импульса τ_2 (цикл T2) выдает на магистрали $\overline{\text{МД0}} - \overline{\text{МД7}}$ данные для считывания задатчиком. По положительному фронту следующего импульса τ_2 (цикл T3) задатчик снимает сигнал $\overline{\text{Ком.ЧТ.ЗУ}}$, а через время t_3 — адрес. Запоминающее устройство через интервал времени t_3 после снятия сигнала $\overline{\text{Ком.ЧТ.ЗУ}}$ освобождает магистраль данных.

Если быстродействие ЗУ не позволяет производить считывание данных задатчиком в соответствии с временной диаграммой (рис. 3.2), то ЗУ по отрицательному фронту сигнала $\overline{\text{Ком.ЧТ.ЗУ}}$ должно сформировать сигнал ГОТОВНОСТЬ длительностью, необходимой для обеспечения цикла считывания информации. Задатчик адреса по сигналу ГОТОВНОСТЬ переходит в режим ожидания, сохраняя на магистрали $\overline{\text{МА0}} - \overline{\text{МА15}}$ установленный адрес, а также сигнал $\overline{\text{Ком.ЧТ.ЗУ}}$ (рис. 3.3). При этом ЗУ по положительному фронту импульса τ_2 (цикл Tn) снимает сигнал ГОТОВНОСТЬ и выдает на магистраль $\overline{\text{МД0}} - \overline{\text{МД7}}$ информацию, которая считывается задатчиком.

Задатчик по положительному фронту импульса τ_2 (цикл Tn + 1) снимает сигнал $\overline{\text{Ком.ЧТ.ЗУ}}$, а через интервал времени t_3 — адрес. ЗУ через интервал времени t_3 после снятия сигнала $\overline{\text{Ком.ЧТ.ЗУ}}$ освобождает магистраль данных.

Запись данных в ЗУ осуществляется в соот-

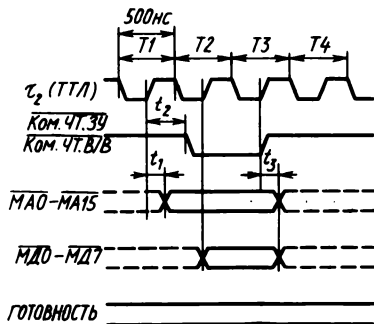


Рис. 3.2. Временная диаграмма чтения данных из памяти

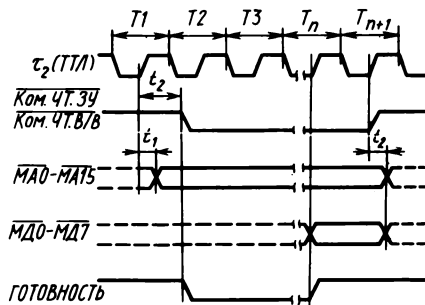


Рис. 3.3. Временная диаграмма чтения данных из памяти с ожиданием сигнала ГОТОВНОСТЬ

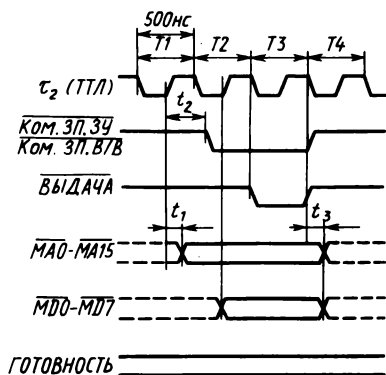


Рис. 3.4. Временная диаграмма записи данных в ЗУ

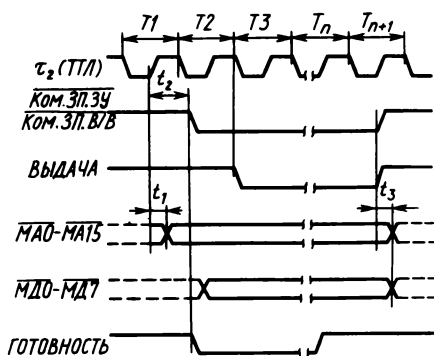


Рис. 3.5. Временная диаграмма записи данных с ЗУ с ожиданием сигнала ГОТОВНОСТЬ

становленного на магистрали данных, используя в качестве строба сигнал ВЫДАЧА.

Если быстродействие ЗУ не позволяет производить запись данных задатчиком в соответствии с временной диаграммой (рис. 3.4), то запись, как и при чтении данных, производится с формированием ЗУ сигнала ГОТОВНОСТЬ длительностью, необходимой для обеспечения цикла записи (рис. 3.5).

Ввод данных из УВВ осуществляется ана-

ветствии с временной диаграммой, представленной на рис. 3.4 и 3.5.

Задатчик, управляющий интерфейсом ИК1, через интервал t_1 после положительного фронта импульса τ_2 (цикл T1) формирует: на магистрали МА0 — МА15 требуемый адрес, через интервал t_2 — сигнал Ком.ЗП.ЗУ, по положительному фронту следующего импульса τ_2 (цикл T2) на магистрали МД0 — МД7 — код информации, предназначенный для записи в ЗУ, а по спаду этого же импульса τ_2 — сигнал ВЫДАЧА. По отрицательному фронту следующего импульса τ_2 (цикл T3) задатчик снимает сигналы Ком.ЗП.ЗУ и ВЫДАЧА, а затем (через интервал времени t_3) адрес и данные. ЗУ по положительному фронту сигнала Ком.ЗП.ЗУ дешифрирует адрес и производит запись кода, ус-

логично чтению данных из памяти, однако при этом задатчик формирует сигнал Ком.ЧТ.В/В, а на магистралях МА0 — МА7 и МА8 — МА15 формируются два одинаковых кода адреса.

Вывод данных в УВВ осуществляется аналогично записи данных в память, однако при этом задатчиком формируется сигнал Ком.ЗП.В/В, а на магистралях МА0 — МА7 и МА8 — МА15 — два одинаковых кода адреса.

Операция прерывания программы инициируется сигналами ВЕКТ. ПЕР.1 — ВЕКТ.ПЕР.8, поступающими в ЦП от периферийных устройств. ЦП воспринимает запросы на прерывание только при условии предварительной программной установки сигнала РАЗР.ПЕР. по выполнению команды EI.

Операции захвата магистрали (прямой доступ) производятся периферийным устройством, выработавшим сигнал ЗАХВАТ и получившим от ЦП сигнал ПОДТВ.ЗАХВ. При этом периферийное устройство становится задатчиком и обеспечивает формирование временных последовательностей в соответствии с рис. 3.2 и 3.5. Предоставляя интерфейс периферийному устройству, ЦП переводит магистрали МА0 — МА15 и МД0 — МД7 в высокоимпедансное состояние. Снимая сигнал ЗАХВАТ в конце операции, задатчик также обеспечивает высокоимпедансное состояние на магистралях адреса МА0 — МА15 и данных МД0 — МД7. После снятия задатчиком сигнала ЗАХВАТ ЦП снимает сигнал ПОДТВ.ЗАХВ. и продолжает выполнение программы.

Магистрали адреса МА0 — МА15 и данных МД8 — МД7 могут находиться в одном из трех состояний: высокоимпедансном, логического 0 и логической 1.

Остальные линии имеют только два состояния: логического 0 и логической 1.

Для подключения к интерфейсу ИК1 используются магистральные формирователи, в качестве которых для передачи сигналов по магистралям МА0 — МА15 и МД0 — МД7 применяются микросхемы К589АП16 и К589АП26.

В качестве магистральных источников для передачи по интерфейсу ИК1 сигналов Ком.ЧТ.ЗУ, Ком.ЧТ.В/В, Ком.ЗП.ЗУ, Ком.ЗП.В/В, ГОТОВНОСТЬ, ВЫДАЧА,

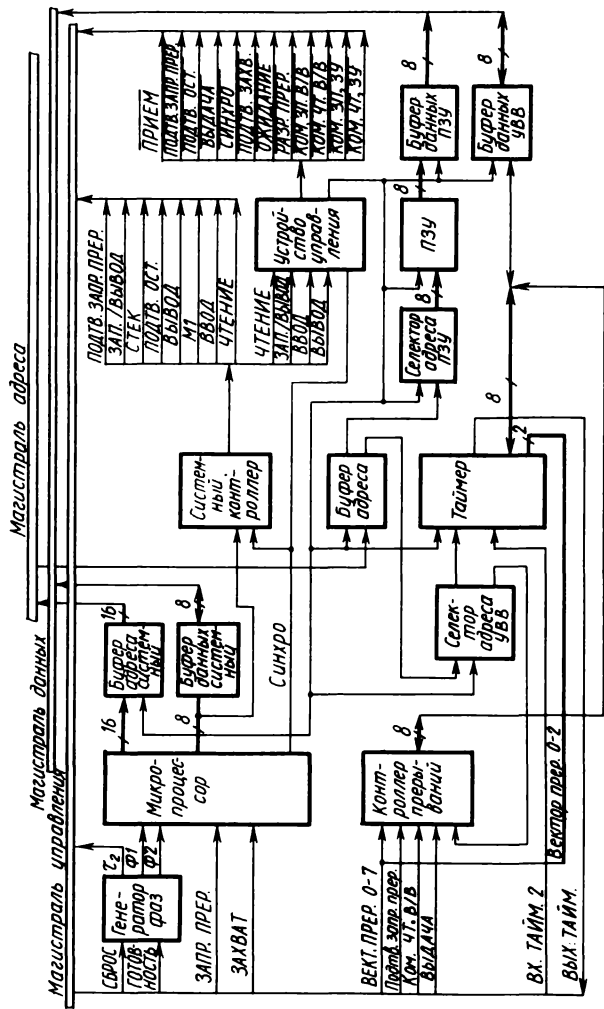


Рис. 3.6. Функциональный состав центрального процессора

ПРИЕМ используются микросхемы К155ЛА7. В качестве источников сигналов ВЕКТ.ПРЕР.1 — ВЕКТ.ПРЕР.8 допускается использование любых микросхем серии К155.

Для присоединения устройств к интерфейсу ИК1 применяются разъемы типа РППМ-17. К магистралям системного разъема интерфейса ИК1 допускается подключение не более четырех приемников или передатчиков.

Центральный процессор. Центральный процессор представляет собой одноплатный контроллер, имеющий в своем составе функциональные узлы, показанные на рис. 3.6. Основой его является микропроцессор КР580ВМ80А, который используется в качестве центрального элемента обработки информации. Обеспечение микропроцессора необходимыми тактирующими сигналами осуществляется с помощью генератора фаз, который формирует сигналы фаз $\Phi 1$ и $\Phi 2$, тактирующие сигналы t_2 , а также управляющие сигналы RESET и \overline{STSTB} , необходимые для работы системного контроллера.

Связь микропроцессора с магистралями адреса $\overline{МА}$ и данных $\overline{МД}$ осуществляется через системные буферы адреса и данных. Буфер адреса обеспечивает однонаправленную передачу информации, буфер данных — двунаправленную. Направление передачи буфера данных определяет сигнал ПРИЕМ, при наличии которого передача информации происходит с магистрали $\overline{МД}$ на выходы микропроцессора.

По сигналу ПОДТВ.ЗАХВАТА, поступающему с системного контроллера, системные буферы адреса и данных устанавливаются в высокоимпедансное состояние. Системный контроллер предназначен для формирования интерфейсных и внутренних управляющих сигналов.

Селектор адреса ПЗУ используется для выборки ПЗУ в области адресов от С000 до FFFF. При наличии сигнала выборки информация с выходов БИС ПЗУ через буфер данных ПЗУ поступает на магистраль данных $\overline{МД0}$ — $\overline{МД7}$. Адресация контроллера прерываний, таймера и устройства управления осуществляется селектором адреса УВВ. При наличии сигнала выборки на одном из перечисленных узлов происходит обмен информацией между магистралью $\overline{МД}$ и выбранным узлом через буфер данных УВВ.

Контроллер прерываний обеспечивает обслуживание восьми запросов на прерывание. По одному из сигналов

ЗАПР. ПРЕР.0—ЗАПР. ПРЕР.7 контроллер прерываний (БИС КР580ВН59) формирует сигнал прерывания INT. При получении сигнала подтверждения прерывания INTA от микропроцессора контроллер выдает на магистраль МД код команды CALL. Микропроцессор, получив этот код, инициирует еще два сигнала INTA, по которым БИС КР580ВН59 выдает на магистраль данных младший и старший байты адреса подпрограммы обслуживания прерывания.

Устройство управления предназначено для управления выбором ПЗУ, а также для формирования внутренних сигналов ЗАПР. ПРЕР.0 — ЗАПР. ПРЕР.2, объединенных с одноименными интерфейсными сигналами монтажной схемой ИЛИ. В связи с тем что ПЗУ и ОЗУ имеют единое поле адресов (от C000 до FFFF), при обращении к одному из них должно блокироваться другое. Сигналы блокировки или выбора устройства, а также сигналы управления формированием запросов на прерывание выдаются устройством управления в зависимости от управляющего слова, поступившего с МД.

Запоминающие устройства. Тип и емкость внутренних запоминающих устройств (ЗУ) микроЭВМ зависят от варианта исполнения. Внутреннее ЗУ может состоять в общем случае из постоянного ЗУ емкостью 16 К байт и оперативного ЗУ динамического типа. ОЗУ микроЭВМ динамического типа, выполненное на основе БИС К565РУЗА, имеет емкость 48 или 64 К байт.

Исполнения микроЭВМ, комплектуемые малой операционной системой МОС 2, имеют ПЗУ емкостью 16 К байт. ПЗУ хранит программы резидентного математического обеспечения и размещается на плате центрального процессора.

Система прерывания. Асинхронная работа внешних устройств и центрального процессора осуществляется в микроЭВМ «Электроника К1-30» с помощью многоуровневой системы прерываний. Прерывающим устройствам в интерфейсе ИК1 отведено восемь линий. Уровень приоритетности запросов может предварительно программироваться в узле приоритетных прерываний (УПП) центрального процессора.

На поступивший запрос УПП вырабатывает синхронное подтверждение прерывания и вектор прерывания. Текущее состояние счетчика команд прерываемой программы записывается в стеке. Возврат к прерванной программе осуществляется с помощью инструкций, ко-

торые должна иметь программа, обслуживающая прерывание. Эта программа, обслуживающая прерывание, в свою очередь, может быть прервана и т. д. Ограничить число прерываний может только глубина стека.

Устройства сопряжения микроЭВМ с периферийными приборами. Аппаратно-программные средства микроЭВМ обеспечивают сопряжение со стандартными периферийными приборами, состав которых для вариантов исполнений приведен в § 3.1.

Устройства сопряжения размещены на трех функциональных модулях: модуле устройства сопряжения с электрифицированной пишущей машинкой «Copsul 260.1», модуле устройства сопряжения с ленточным перфатором ПЛ-150 с фотосчитывающим устройством FS-1501; модуле сопряжения с внешними устройствами. При этом модуль сопряжения с внешними устройствами обеспечивает обмен информацией с видеотерминалами алфавитно-цифровым ВТА2000-2/1, 15ИЭ200 × 140-017, накопителями на гибких магнитных дисках «Электроника ГМД-70», 15ВВМД-512-002, «Электроника ГМД-7012», устройствами печати, имеющими интерфейс типа ИППР (термопечатающие устройства, УВВ ПЧ30-004, DAR01156, DZM180 и др.). Устройства сопряжения обеспечивают работу микроЭВМ с периферийными приборами по флаговому принципу. При выходе на драйвер обмена с конкретным внешним устройством микроЭВМ опрашивает бит состояния ГОТОВНОСТЬ ВВОДА или ГОТОВНОСТЬ ВЫВОДА соответствующего устройства сопряжения до установления его действительного значения, после чего выполняются команды ВЫВОД или ВВОД.

Режимы работы микроЭВМ. В микроЭВМ реализуются автоматический командный и шаговый режимы работы. Обеспечение указанных режимов обусловлено возможностью управления сигналом ГОТОВНОСТЬ микропроцессора. Устанавливая на входе ГОТОВНОСТЬ напряжение низкого уровня, приостанавливают работу микропроцессора, который переходит в состояние ожидания. Когда на магистрали данных МД0 — МД7 появляется информация, готовая для приема в микропроцессор, на входе ГОТОВНОСТЬ устанавливается напряжение высокого уровня и микропроцессор продолжает работу по программе.

Выполнение программы в автоматическом режиме предусматривает постоянное наличие на выходе пульта управления сигнала ГОТОВНОСТЬ высокого уровня.

Выполнение программы в шаговом режиме позволяет локализовать аппаратные и программные ошибки. В отладочных режимах работы микроЭВМ (командный, шаговый) сигналы микропроцессора СИНХРО и М1 управляют формированием сигнала ГОТОВНОСТЬ. С помощью сигнала СИНХРО реализуется шаговый режим работы, т. е. выполнение каждой команды осуществляется по машинным циклам. С помощью сигнала М1, СИНХРО реализуется командный режим работы, т. е. выполнение программы происходит в последовательности команда за командой.

Аппаратура, предназначенная для организации указанных режимов работы микроЭВМ, расположена в устройстве сопряжения с пультом управления.

Контроль микроЭВМ. Проверка работоспособности микроЭВМ осуществляется с помощью комплекта программ, входящих в состав системных программных средств. Комплекты программ обеспечивают: проверку микроЭВМ с локализацией места неисправности с точностью до модуля; использование контролирующих программ при ручном поиске неисправного логического элемента.

Системные программные средства контроля состоят из контролирующих программ центрального процессора, ОЗУ, устройства сопряжения с ЭПМ «Consul 260.1», устройства сопряжения с FS-1501 и ПЛ-150; устройств ввода — вывода; устройства сопряжения с накопителем «Электроника ГМД-70» 15ВВМД-512-002, видеотерминального устройства ВТА2000-2/1, 15ИЭ200 × 140-017 и устройства печати.

Контролирующие программы микроЭВМ построены по принципу расширяющихся областей. Этот принцип заключается в том, что в выполнении очередной тестовой программы участвует аппаратура, работоспособность которой проверялась предыдущим тестом. В зависимости от задачи контроля отдельные контролирующие программы могут объединяться и выполняться в произвольной последовательности.

При проверке работоспособности микроЭВМ контролирующая программа выполняется под управлением монитора с отображением результатов проверки на выбранном устройстве вывода. При этом программа с внешнего носителя записывается в ОЗУ, после чего этой программе автоматически или с пульта передается управление.

3.3. Конструктивное исполнение микроЭВМ «Электроника К1-30»

Все устройства микроЭВМ, за исключением пульта управления (ПУ), выполнены на печатных платах размером 200×300 мм. На сторонах плат размером 300 мм расположены печатные разъемы, предназначенные для соединения устройств посредством розеток РППМ-17 в систему и с периферийными устройствами. Платы имеют двусторонний печатный монтаж. Питание к интегральным схемам подводится с помощью навесных шин. Ламели выходных печатных разъемов имеют износостойкое антикоррозионное покрытие с малым удельным сопротивлением.

Все устройства микроЭВМ с блоками питания смонтированы в блок (рис. 3.7), устанавливаемый в шкаф-стойку либо используемый в настольном варианте. Габаритные размеры его $550 \times 483 \times 133$ мм. Несущим узлом блока является шасси 6, в передней части которого имеется рамка 8, предназначенная для установки в ней ПУ, а также для закрепления блока в стойке.

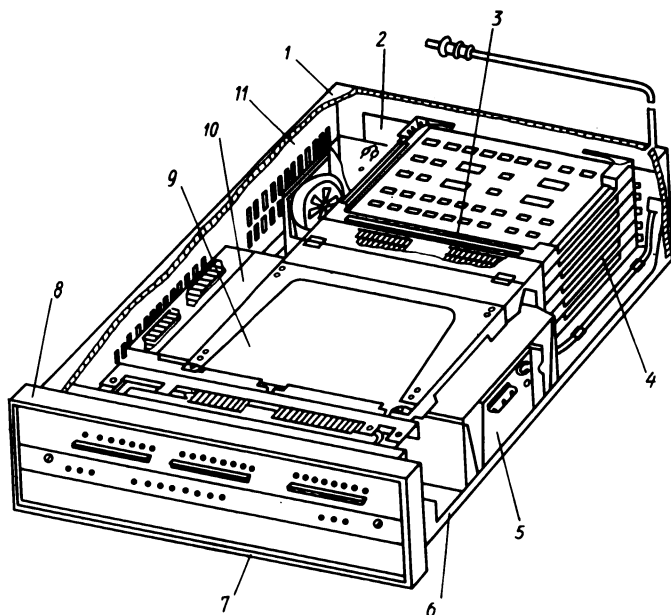


Рис. 3.7. Конструкция микроЭВМ «Электроника К1-30»

К шасси крепятся все основные узлы и детали микроЭВМ. В передней части устанавливается блок питания 5, который крепится при помощи перегородок к шасси и закрывается крышкой, являющейся электрическим экраном и одновременно воздушным каналом для продувки отсека 10 блока питания.

Для установки устройств 4 микроЭВМ используется кассета, представляющая собой П-образную жесткую конструкцию, в которой имеются направляющие и места для установки системных разъемов 3. В рабочем положении кассета закрепляется таким образом, что шесть устройств, установленных в ней, своими интерфейсными разъемами обращены к задней части блока, а устройство сопряжения с пультом управления — к пулту управления.

В задней части установлена панель 2 с окнами для кабельных розеток, которые после сочленения с вилками интерфейсных разъемов крепятся к задней панели.

Пульт управления 7 крепится внутри рамки и своими разъемами сочленяется с интерфейсными разъемами устройства сопряжения 9 с ПУ. Пульт состоит из штампованной панели, на которой закреплены блоки переключателей П2К, платы со светодиодными индикаторами, а также угольник с розетками РППМ-17.

Органы управления расположены на пульте группами в два ряда: в верхнем ряду — 16 переключателей адреса и 8 переключателей данных; симметрично клавишам сверху имеются окна индикации; посередине — устройство индикации состояния микропроцессора; нижний ряд переключателей кроме оставшихся органов управления имеет два переключателя СЕТЬ-ВЫКЛ. и НАГРУЗКА — ОТКЛ, вынесенных из блока питания. Органы управления дают возможность использования ПУ в качестве пульта оператора при отладке математического обеспечения и в качестве инженерного пульта при наладочных работах. При этом пульт обеспечивает: включение и отключение электропитания; установку микроЭВМ в исходное состояние; пуск и останов микроЭВМ; задание — автоматического, командного, шагового, отладочного и автоматического с остановом при сравнении адресов — режимов работы микроЭВМ; автоматическое наращивание адресов при обращении к ЗУ; набор кодов адреса и данных; чтение — запись информации из (в) ЗУ и УВВ; индикацию кодов адреса, данных управляющих сигналов микропроцессора и напряжения вторичных

источников питания; имитацию восьми внешних запросов на прерывание.

В задней части шасси микроЭВМ имеются розетка для подключения ЭПМ «Consul 260.1», клемма КОРПУС и окно для выхода сетевого кабеля блока питания. К шасси крепится кожух 1, опирающийся в передней части на приливы рамки, а в задней — на заднюю панель.

Тепловые режимы обеспечиваются принудительной приточно-локальной системой воздушного охлаждения 11. Отсек, в котором установлен блок питания, образует первый отдельный воздушный канал, продуваемый двумя вентиляторами ВВФ-71М. Все остальные устройства микроЭВМ образуют второй воздушный канал, поток воздуха в котором создается вентилятором ВВФ-71, установленным на шасси против окон между платами.

Монтаж микроЭВМ по системному разъему выполнен проводом МНВ 0,12 мм² методом накрутки. При эксплуатации микроЭВМ в настольном варианте шасси устанавливается на опорные амортизаторы АО-30.

3.4. Контроллер программируемый универсальный «Электроника К1-20»

Контроллер программируемый универсальный (КПУ) «Электроника К1-20» предназначен для использования в составе систем управления технологическим оборудованием, а также в составе испытательного и контрольно-измерительного оборудования.

В соответствии с основным назначением КПУ конструктивно выполнен в виде собственно контроллера и пульта управления, причем в зависимости от варианта исполнения КПУ имеют различные составы, емкости и типы программируемых и электрически перепрограммируемых ПЗУ. Контроллер выполнен на основе БИС серий К580, К589, К565, КР556, К573, К567 и К155. Базовым комплектом являются БИС серии К580.

Основные технические характеристики КПУ следующие:

Разрядность — восемь двоичных разрядов

Форма представления чисел — двоичный код с фиксированной точкой; имеется возможность выполнения арифметических операций и ряда тригонометрических, показательных и логарифмических функций с плавающей точкой при помощи библиотеки подпрограмм

Виды операции — неявная, прямая, косвенная, непосредственная

Прямая адресация к памяти — до 64 К байт

Максимальное число адресуемых ВУ — 256 устройств ввода и 256 устройств вывода

Система команд — аналогична микроЭВМ «Электроника К1-30»

Система прерываний — приоритетная с программной установкой приоритета, режимов обслуживания и масок, с шестью запросами на прерывание при работе с ПУ и восемью — без ПУ

Емкость ОЗУ — 1 К байт

Емкость ПЗУ, ПППЗУ и ЭППЗУ — переменный параметр, зависящий от варианта КПУ

Общее число программируемых параллельных каналов ввода — вывода — 48, в том числе имеющих гальваническую развязку — 8 (4 — для ввода, 4 — для вывода)

Число программируемых последовательных каналов ввода информации — 1, вывода — 1

Число 16-разрядных программируемых интервальных таймеров — два при работе с ПУ и три при работе без ПУ

Обмен информацией между ПУ и контроллером — программно-аппаратный в режиме реального времени

Отладка программ пользователя — с применением управляющей программы МОНИТОР

Контроль работоспособности КПУ — с помощью тест-программ

Ручной ввод информации — с 16 клавиш ПУ

Ввод операции управления — с 3 клавиш ПУ

Число позиций индицируемых символов — 9

Режим эксплуатации КПУ — непрерывный или периодический с многократным включением (выключением)

Возможно расширение ЗУ — до 64 К байт; устройств ввода — вывода — до 256

Возможно перераспределение зон памяти контроллера и дополнительных ЗУ, подключаемых пользователем

Режим работы КПУ — автоматический. При отладке программ имеется возможность работы в командном режиме

Установка КПУ и систем пользователя в исходное состояние — вручную клавишей Р (СБРОС) на ПУ и автоматически при включении питания

Габаритные размеры корпуса контроллера — не более 358×244×57 мм, ПУ — 237×160×40 мм

Масса контроллера — не более 1,5 кг, ПУ — 0,8 кг

Потребляемая мощность КПУ — не более 35 Вт

Номиналы питающих напряжений: +5; +12; —5 В; токов: 5; 0,4; 0,1 А соответственно

Из изложенного видно, что КПУ — весьма эффективное средство отладки программного обеспечения непосредственно на объекте пользователя. Наличие пульта управления, обеспечивающего покомандную отладку, и имеющаяся возможность последовательной отработки программного обеспечения в ЭППЗУ (К537РФ2), ПППЗУ

(КР556РТ5) и ПЗУ (К568РЕ1) обуславливают гибкость КПУ по отношению к изменениям требований к программному обеспечению в процессе разработки и позволяют существенно сократить этап проектирования.

Структурные особенности, интерфейс и система команд КПУ определяются типом примененного микропроцессора. БИС КР580ВМ80А является центральным элементом обработки информации в КПУ.

Характеристики структуры и системы команд КПУ в значительной степени соответствуют аналогичным характеристикам микроЭВМ «Электроника К1-10», «Электроника К1-30», здесь же приведем лишь отличия с точки зрения пользователя.

Интерфейс пользователя представляет собой совокупность линий разъемов контроллера ХС2 и ХС3. Он обеспечивает программируемый побайтовый обмен информацией контроллера с устройствами пользователя по 48 линиям (в том числе 4 входа и 4 выхода с гальванической развязкой); программируемый прием информации по одному каналу в последовательном коде; программируемую передачу информации по одному каналу в последовательном коде; формирование программно-управляемых временных задержек с выходов трех интервальных таймеров без ПУ; работу устройств системы пользователя с контроллером в режиме прерываний; подачу из системы пользователя в контроллер напряжений питания.

Кроме того, пользователь может менять направление обмена информации с помощью объемных перемычек в коммутационном поле в зависимости от конкретного типа устройства и протокола обмена.

Объекты пользователя подсоединяются к интерфейсу при помощи вилок СНПЗУ-69/132×12,5 В-21.

Программное обеспечение КПУ предоставляет пользователю средства контроля и отладки алгоритмов программ в режиме реального времени, позволяет снизить трудоемкость и повысить качество программирования алгоритмов, реализуемых КПУ. В программное обеспечение входят управляющая программа диалогового взаимодействия пользователя с КПУ — МОНИТОР КПУ; библиотека подпрограмм обработки чисел с плавающей точкой; тест-программа работоспособности КПУ.

Функциональные части КПУ (контроллер и ПУ) выполнены в виде самостоятельных конструктивов. Для связи контроллера с устройствами управления, устройст-

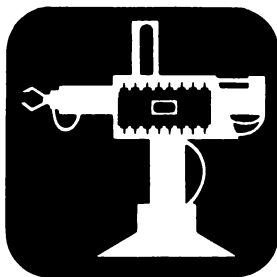
вами расширения системы и пультовыми устройствами применен разъем типа СНП-34. Розетки установлены на длинной стороне платы контроллера. В местах расположения розеток СНП-34 в корпусе предусмотрены окна для присоединения кабельных вилок СНП.

Пульт управления в процессе эксплуатации может располагаться на столе или в руке оператора. Дисплейная линейка пульта собрана на цифровых индикаторах АЛС324А; клавиатура пульта — на основе клавиш блока ВМ-27-1-1.

Большие интегральные схемы К580 и БИС ЗУ устанавливаются на специальных колодках, что обеспечивает удобство при контроле и ремонте КПУ. На контактных колодках устанавливаются также перепрограммируемые БИС ЗУ, так как пользователь должен программировать ЗУ самостоятельно. Кроме того, пользователю предоставлено право выбора и смены типа ЗУ, что обеспечивается применением конструктивных элементов для перекоммутации связей.

Плата контроллера расположена в корпусе, собранном из двух одинаковых крышек. Крышка имеет шесть бобышек для установки платы контроллера и четыре бобышки с отверстиями, через которые обе крышки скрепляются между собой.

Однокристалльные микроЭВМ серии K1816



Рассматриваемое семейство представляет собой однокристалльные 8-разрядные микроЭВМ серии K1816 — KM1816BE48, KP1816BE39, KP1816BE49, предназначенные для применения в устройствах различного назначения (от локальных систем автоматики до устройств управления бытовыми приборами), использования на нижних уровнях систем управления и в разнообразных высокопроизводительных системах управления и обработки данных.

4.1. Назначение, структура микроЭВМ семейства, состав функциональных узлов

Микросхемы семейства — функционально законченные устройства, содержащие центральный процессор, ОЗУ данных и ЗУ программ, многоканальной интерфейс ввода — вывода, 8-разрядный таймер-счетчик, векторную схему прерываний, тактовый генератор, устройство синхронизации. Микросхемы имеют идентичную структуру и отличаются быстродействием, типом и объемом внутренней памяти программ, объемом внутреннего ОЗУ [7, 8].

В табл. 4.1 приведены отличительные особенности микросхем серии K1816.

В каждой схеме предусмотрена возможность расширения памяти программ по 4 К байт, памяти данных до 384 байт и числа линий ввода — вывода за счет подключения внешней памяти программ (ПЗУ), ОЗУ и интерфейсов ввода — вывода серии KP580.

Конструктивно микросхемы выполнены в металло-керамическом (KM1816BE48) и пластмассовых (для KP1816BE39, KP1816BE49) 40-выводных корпусах с двухрядным расположением выводов (микросхема KM1816BE48 с прозрачной для ультрафиолетового из-

Таблица 4.1

Микросхема	Тип программной памяти	Объем памяти программ ПЗУ, байт	Объем памяти данных ОЗУ, байт	Максимальная тактовая частота, МГц
KP1816BE39	Нет	Нет	128	11,0
KM1816BE48	ППЗУ со стиранием ультрафиолетовым излучением	1К	64	6,0
KP1816BE49	ПЗУ	2К	128	11,0

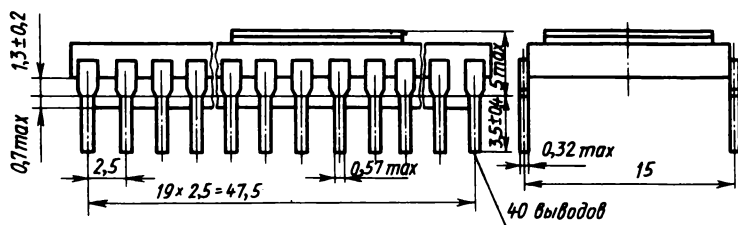


Рис. 4.1. Конструктивное исполнение БИС KM1816BE48

лучения крышкой (рис. 4.1, 4.2). Назначение выводов приведено в табл. 4.2.

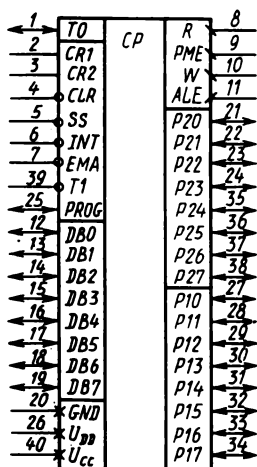


Рис. 4.2. Условное графическое обозначение

МикроЭВМ состоит из следующих функциональных узлов (рис. 4.3): арифметическо-логического устройства (АЛУ), блока программной памяти (БПП), счетчика команд (СК), ОЗУ, устройства управления и синхронизации, таймера-счетчика, схемы прерывания и регистра состояния программы.

Арифметическо-логическое устройство — 8-разрядное, позволяет выполнять арифметические, логические операции и операции сдвига над данными. В свою очередь, АЛУ содержит собственно комбинационную схему арифметическо-логического устройства, аккумулятор, регистр аккумулятора, регистр временного

Таблица 4.2

Номер вывода	Обозначение вывода	Назначение вывода
1	T0	Тестируемый вход 0
2	CR1	Вход тактового генератора
3	CR2	Выход тактового генератора
4	$\overline{\text{CLR}}$	Сброс
5	$\overline{\text{SS}}$	Пошаговый режим
6	$\overline{\text{INT}}$	Прерывание
7	$\overline{\text{EMA}}$	Выборка из внешней памяти
8	$\overline{\text{R}}$	Чтение
9	PME	Считывание из внешней памяти
10	$\overline{\text{W}}$	Запись
11	ALE	Фиксация адреса
12—19	DB0—DB7	Магистраль данных (Порт P0)
20	GND	Общий
21—24, 35—38	P20—P27	Порт P2
25	PROG	Строб расширителя ввода — вывода
26	U_{DD}	Напряжение питания при программировании
27—34	P10—P17	Порт P1
39	$\overline{\text{TI}}$	Тестируемый вход 1
40	U_{CC}	Напряжение источника питания + 5 В

хранения, схему десятичной коррекции аккумулятора. Аккумулятор используется для записи и хранения результата выполнения операции АЛУ; регистр временного хранения — для фиксации второго операнда, участвующего в операции АЛУ; регистр аккумулятора — для первого операнда; схема десятичной коррекции имеет то же функциональное назначение, что и в микропроцессорах КР580 и К1810.

Блок программной памяти предназначен для хранения, считывания (а в микроЭВМ КМ1816ВЕ48 — и для записи) команд, которые поступают в центральный процессор и управляют обработкой информации. БПП состоит из ПЗУ (или ППЗУ), счетчика команд, дешифратора адреса и дешифратора команд.

Память, расположенная на кристалле, занимает адреса 000H — 03FFH (КМ1816ВЕ48) или 07FFH (КР1816ВЕ49) (рис. 4.4). Внешняя память занимает адреса с 0400H (КМ1816ВЕ48) или с 0800H (КР1816ВЕ49) по 0FFFH. Все поле адресов 000H — 0FFFH разбито на два банка:

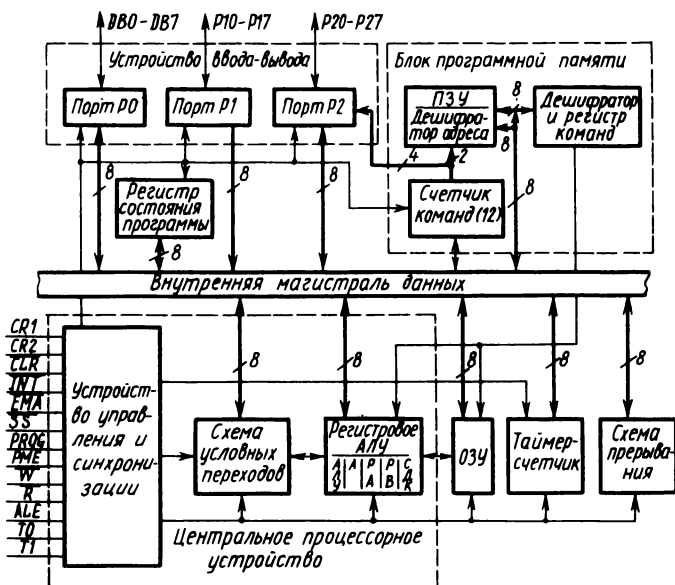


Рис. 4.3. Структура микроЭВМ

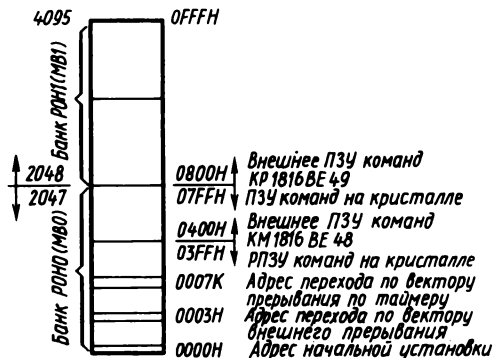


Рис. 4.4. Распределение поля адресов программной памяти

банк 0 с адресами 000H — 07FFH и банк 1 с адресами 0800H — 0FFFH. Переключение банков осуществляется программным путем.

Для формирования текущего адреса команды в программной памяти применен 12-разрядный счетчик команд

(СК). Содержимое СК последовательно увеличивается после выбора каждого байта команды и изменяется скачкообразно при выполнении команд условных, безусловных переходов и при прерываниях. Старший разряд СК изменяется только программно (команды SEL MB0, SEL MB1). СК разбит на две части: счетчик младших разрядов (биты 0—7) и счетчик старших разрядов (биты 8—11). При использовании внешней программной памяти биты 0—7 поступают через порт P0 (DB0—DB7), а биты 8—11 — через порт P2 (P20 — P23).

Декодирование команд, поступающих из программной памяти, осуществляется с помощью регистра и дешифратора команд (ДК). Регистр используется для записи и хранения кодов команд, передаваемых через внутреннюю шину с выхода ПЗУ (ППЗУ) или из порта P0 при внешнем ПЗУ. Дешифратор команд представляет собой программируемую логическую матрицу, на вход которой поступает код команды, а с выхода снимаются управляющие сигналы, выполняющие эту команду.

Оперативное запоминающее устройство данных разбито на два банка РОН: банк 0 с адресами 00H—07H и банк 1 с адресами 18H—1FH. Переключение банков осуществляется программным способом командами SEL RB0, SEL RB1. Восьмиуровневый 16-рядный стек с адресами 08—17 и ячейки ОЗУ 20—3FH используются только как ОЗУ данных (рис. 4.5).

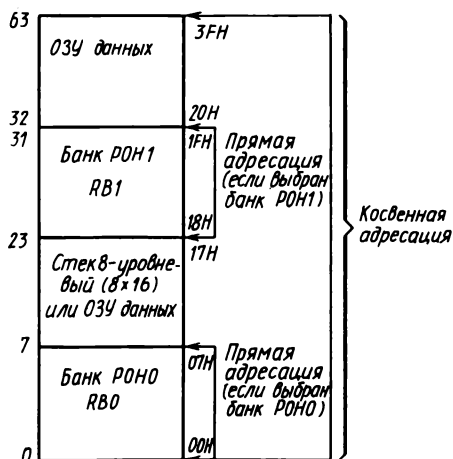


Рис. 4.5. Распределение адресов ОЗУ

В микроЭВМ предусмотрена возможность расширения памяти данных путем подключения микросхем ОЗУ емкостью до 256 байт. Общая емкость ОЗУ в этом случае равна 324 байт. Обращение к внешней памяти возможно с помощью команд $\text{MOVX } \bar{A}, R, A$; $\text{MOVX } A, \bar{R}$. Стробится информация внешнего ОЗУ сигналами \bar{W} , \bar{R} . В микроЭВМ имеется 27 линий ввода — вывода, из которых 24 линии объединены в три 8-разрядных порта (P0, P1, P2). Порт P0 (DB0—DB7) — двунаправленный, порты P1, P2 — квазидвунаправленные. Порты P1 и P2 имеют идентичные характеристики. Данные, записанные на них, статически фиксируются и не изменяются до перезаписи. Как входы эти линии не фиксируются, т. е. входные данные должны присутствовать до считывания по команде приема данных.

Для использования портов P1, P2 в качестве входов микроЭВМ должна выдать 1 на соответствующие линии P1, P2 по команде выдачи данных. Порты P1, P2 устанавливаются в состояние 1 также после подачи сигнала CLR . Порт P0 — 8-разрядный двунаправленный порт с тремя состояниями — может использоваться в качестве статически фиксированного выходного порта или нефиксированного входного порта. Информация, выдаваемая портом P0 с помощью команд $\text{OUTL BUS}, A$ и $\text{MOVX } \bar{A}, R, A$, сопровождается строб-импульсом \bar{W} . При записи информации в порт P0 с помощью команд INSA, BUS и $\text{MOVX } A, \bar{R}$ вырабатывается строб-импульс \bar{R} . Кроме операций ввода — вывода информации предусмотрена возможность выполнения логических операций И, ИЛИ непосредственно на портах P0, P1 и P2 с помощью команд $\text{ANL P}, \# \text{DATA}$; $\text{ORL P}, \# \text{DATA}$; $\text{ANL BUS}, \# \text{DATA}$; $\text{ORL BUS}, \# \text{DATA}$.

Три линии ввода — вывода T0, T1, INT служат входами, проверяемыми командами условного перехода. Линия ввода — вывода T0 используется для выдачи тактовых сигналов, линия T1 — как вход счетчика внешних событий для таймера счетчика, линия INT — для внешнего аппаратного прерывания.

Число линий ввода — вывода можно увеличивать, если использовать команды $\text{MOVD } A, P$; $\text{MOVD } P, A$; $\text{ANLD } P, A$; $\text{ORLD } P, A$ и соответствующую схему. При этом обмен информацией осуществляется через порт P2 (P20—P23) (рис. 4.6). Каждая пересылка состоит из двух 4-разрядных полубайтов: первый полубайт со-

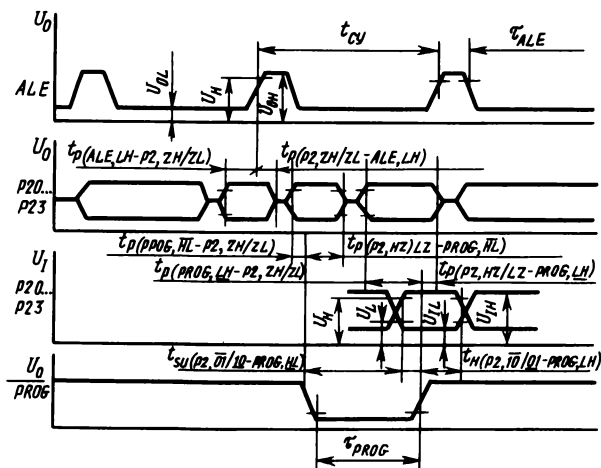


Рис. 4.6. Временная диаграмма работы с дополнительным портом

держит код операции и адрес порта, второй — 4 бит данных А. Синхронизация осуществляется сигналом PROG. Переход сигнала PROG из состояния высокого уровня в состояние низкого уровня означает, что на выводах P20—P23 находятся код операции и адреса порта, а обратный переход означает, что на этих выводах находятся данные — содержимое четырех младших разрядов аккумулятора.

Устройство управления и синхронизации состоит из генератора, формирователя внутренних тактовых сигналов, формирователей сигналов состояний и режимов работы. Встроенный генератор — последовательная резонансная схема, работающая в диапазоне 1—6 МГц для КМ1816ВЕ48 и 1—11 МГц для КР1816ВЕ39, КР1816ВЕ49. Выводы CR1, CR2 используются для подключения внешнего кварцевого резонатора и LC-цепи (рис. 4.7), которые обеспечивают частотно-зависимую обратную связь и фазовый сдвиг для генератора. На выводы CR1 и CR2 можно подавать также сигналы от внешнего источника тактовых сигналов.

Формирователь внутренних тактовых сигналов (рис. 4.8) делит частоту встроенного генератора на три и вырабатывает внутренние тактирующие сигналы, выдаваемые на вывод T0 по команде ENTO CLK. С выхода делителя

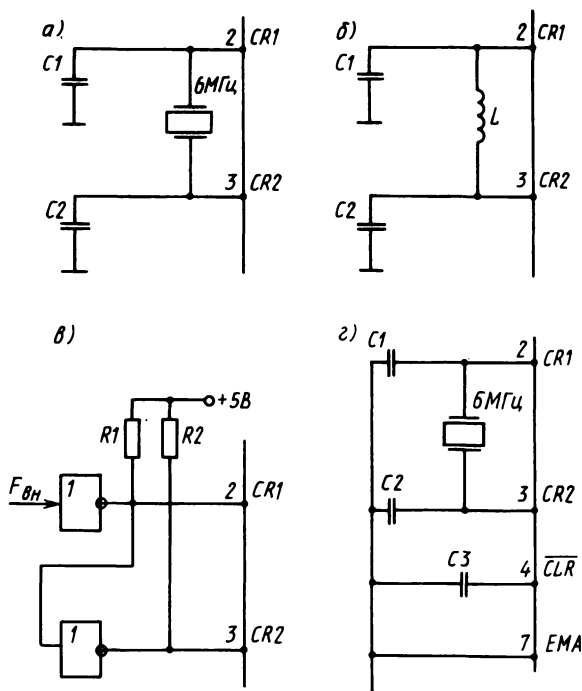


Рис. 4.7. Схемы подключения выводов CR1 и CR2 встроенного генератора:
а — г — варианты схем

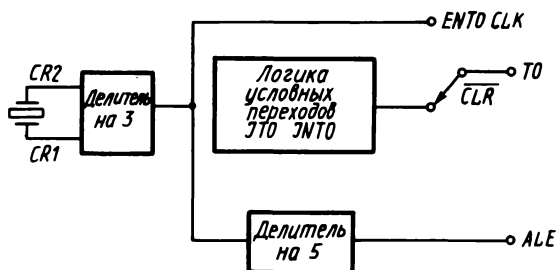


Рис. 4.8. Схема формирования внутренних тактовых сигналов

тактирующие сигналы поступают на вход делителя на 5, выходные сигналы которого определяют пять внутренних состояний микроЭВМ (S1—S5). Сигнал с одного из выходов делителя поступает на вывод ALE. Период следования сигнала ALE соответствует машинному циклу микроЭВМ. Формирователи сигналов состояний и режимов работы определяют режимы работы микроЭВМ. В качестве входных сигналов в формирователях используются сигналы EMA, PROG, TO, \overline{SS} , \overline{CLR} . Их комбинации позволяют реализовать основные режимы работы микроЭВМ.

Схема условных переходов предназначена для формирования сигналов управления ветвлением программы при выполнении команд условных переходов. Переход осуществляется по содержимому аккумулятора (0 или не 0) и по состояниям одного из разрядов аккумулятора: триггера флага таймера-счетчика TT, входов T0, T1, \overline{INT} , триггеров флага F0 и F1, триггера переноса C. Условия перехода определяются соответствующими командами. Значения признаков C, F0, F1, TT устанавливаются программно, значение признаков T0, T1, \overline{INT} — аппаратно.

Таймер-счетчик предназначен для подсчета внешних событий (используется внешний сигнал T1) и генерирования временных интервалов. Он состоит из делителя на 32, счетчика, триггера флага (рис. 4.9). На вход счетчика поступают импульсы с делителя на 32 частотой $F_{CR}/480$ (в режиме таймера) и с входа T1 (в режиме счетчика внешних событий). Содержимое счетчика устанавливается по команде MOV T, A, а текущее значение проверяется по команде MOV A, T. Запуск счетчика в режиме таймера осуществляется командой STRT T, а

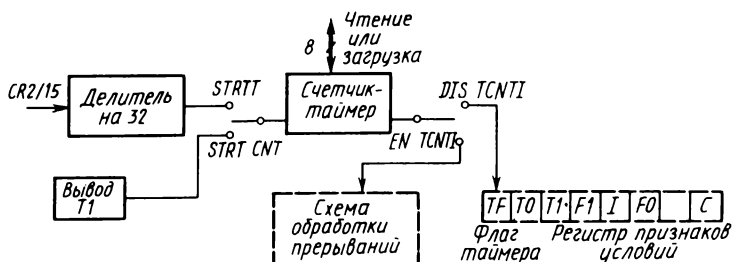


Рис. 4.9. Структура таймера-счетчика

в режиме счетчика событий — командой **STRT CNT**. Останов счетчика производится командой **STOP TCNT**.

Триггер флага таймера-счетчика устанавливается в состояние 1 при переходе счетчика из состояния FF в состояние 00. Триггер используется в командах условного перехода, а также для организации прерывания по таймеру-счетчику.

В микроЭВМ предусмотрено два вектора прерывания: один — аппаратный по входу **INT** (адрес 03H), другой используется как внутренний и как внешний (вход T1) по флагу таймера-счетчика (адрес 07H). Каждый из векторов прерываний может быть разрешен или запрещен программно командами **EN I**, **DIS I**, **EN TCNTI**, **DIS TCNTI**. Кроме того, по сигналу **CLR** прерывание запрещается до тех пор, пока оно не будет разрешено соответствующими командами (**EN I**, **ENI TCNTI**).

При поступлении на вход сигнала **INT** (0 активный), если прерывание по входу **INT** разрешено, происходит обращение к программе обслуживания с адресом 03H. Как и при любом обращении к подпрограмме, содержимое счетчика команд и слово с информацией о состоянии программы (**PSW**) записываются в стек. То же происходит и при выработке флага таймера-счетчика. Однако в этом случае обращение производится к программе обслуживания по адресу 07H. Программа обслуживания должна заканчиваться командой **RETR** (восстановление состояния и возврат к прерванной программе), так как до выполнения этой команды последующие запросы на прерывание не обслуживаются (прерывание запрещено). Система прерывания в микроЭВМ — фиксированная по приоритету. Прерывание по входу **INT** имеет старший приоритет.

Регистр состояния программы **PSW** предназначен для хранения данных о состоянии микроЭВМ. Назначение разрядов **PSW**: 0—2 — разряды указателя стека (**S0—S2**); 3 — разряд не используется (при чтении всегда 1); 4 — разряд указывает на используемый банк рабочих **РОН**; 5 — разряд — флаг пользователя (**F0**), используемый по команде условного перехода; 6 — разряд дополнительного переноса (**AC**), используемый для десятичной коррекции; 7 — разряд — перенос, указывающий на переполнение аккумулятора после предыдущей операции (**CU**). Регистр **PSW** может программно проверяться, модифицироваться полностью и поразрядно.

4.2. Основные режимы работы

МикроЭВМ может находиться в следующих режимах: проверки программной памяти, работы с внутренней и внешней памятью, пошагового выполнения команд, программирования внутренней памяти программы (для КМ1816ВЕ48). Режим работы устанавливается комбинацией входных и выходных сигналов. Инициализация микросхемы осуществляется сигналом \overline{CLR} (активный — низкий уровень напряжения). Этот сигнал выполняет следующие функции: устанавливает счетчик команд и указатель стека (в PSW) в 0; выбирает банк РОН0 (PB0) и банк внутренней программной памяти (MB0); устанавливает порт P0 в высокоимпедансное состояние (при $EA=0$); подготавливает порты P1 и P2 для приема информации; блокирует прерывание по выходу \overline{INT} и таймеру-счетчику; останавливает таймер-счетчик; устанавливает флаги F0 и F1 в 0; запрещает выдачу импульсов по выводу T0. В режиме проверки программной памяти (рис. 4.10) контролируется правильность хранящейся в памяти информации, записанной при программировании, а также «чистота» памяти после стирания (только для БИС КМ1816ВЕ48). Подаваемые на одноименные выходы микросхемы сигналы выполняют следующие функции: \overline{EMA} активизирует режим обращения к внутренней памяти для программирования при подаче напряжения

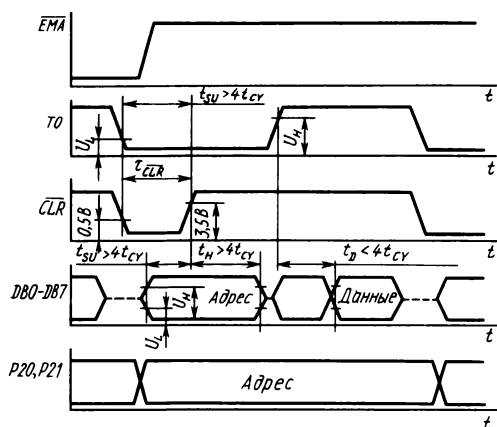


Рис. 4.10. Временные диаграммы режима проверки программной памяти

высокого уровня; T0 обеспечивает режимы программирования (низкий уровень напряжения) и контроля (высокий уровень напряжения); $\overline{\text{CLR}}$ фиксирует выбранный адрес; DB0—DB7 организуют подачу адреса A0—A7 и данных; P20, P21 обеспечивают подачу адреса A8, A9.

Режим контроля памяти организуется таким образом, что на выводы DB0—DB7 подаются адресные сигналы и выдаются данные для контроля. Поэтому при переходе к режиму контроля необходимо обеспечить высокоимпедансное состояние на данных выводах, которое исключает попадание на открытые выводы схемы напряжения адресных сигналов, поступающих от источника.

Режим работы с внутренней памятью устанавливается заданием низкого уровня напряжения на выводе $\overline{\text{ЕМА}}$. Выполнение программы, хранящейся в памяти, начинается после ее инициализации с команды, расположенной по адресу 00H, так как счетчик команд по сигналу $\overline{\text{CLR}} = 0$ сбрасывается в нуль.

Для увеличения производительности в микроЭВМ предусмотрено совмещение выполнения внутренних операций в одном цикле. Например, выполнение выбранной команды и подготовка следующего адреса команды происходит одновременно.

Режим работы с внешней памятью (рис. 4.11) (для КМ1816ВЕ48 и КР1816ВЕ49) применяется при отладке программ и контроле процессора микроЭВМ. При этом внутренняя память отключается подачей на вывод $\overline{\text{ЕМА}}$ напряжения +5 В и используется внешняя память (например, ОЗУ или ППЗУ). Этот режим работы используется также в том случае, если внутренней памяти команд недостаточно — можно или совместить внутреннюю и внешнюю память команд и получить общий объем до 4096 байт (на вывод $\overline{\text{ЕМА}}$ подано напряжение низкого уровня), или использовать только внешнюю память с объемом до 4096 байт (вывод $\overline{\text{ЕМА}}$ подключается к источнику напряжения +5 В). Для БИС КР1816ВЕ39 используется только внешняя память команд. При работе с внешней памятью выдача младших разрядов адреса (A7—A0) на внешнюю память и прием кода команд из внешней памяти осуществляютс^я через порт P0 (DB7—DB0). При этом адрес ЗУ фикси-

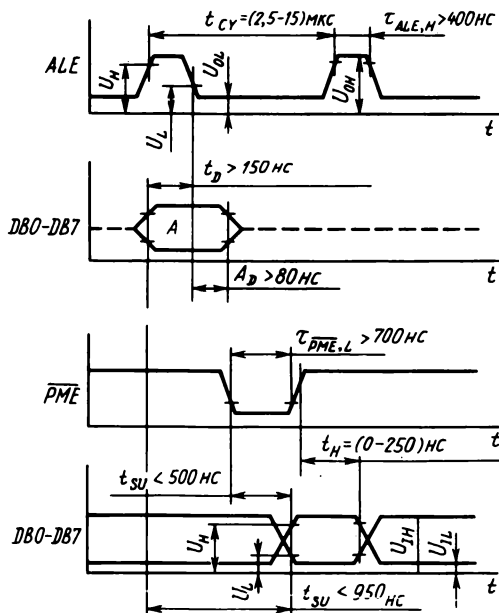


Рис. 4.11. Временные диаграммы режима работы с внешней памятью

руется по сигналу ALE; принимаются команды в микроЭВМ по сигналу PME. Старшие разряды адреса A11 — A8 выдаются через порт P2 (P23—P20).

При увеличении объема памяти данных за счет подключения внешнего ОЗУ емкостью до 256 байт обмен данными между ОЗУ и микроЭВМ осуществляется через двунаправленный порт P0 с помощью команд MOVX A, @ R; MOVX @ R, A (рис. 4.12, а и б). При этом адрес данных фиксируется по сигналу ALE, а прием и выдача данных — по сигналам R и W.

Режим пошагового выполнения программы (рис. 4.13) служит для отладки и проверки программы. В качестве управляющего входа микроЭВМ используются вывод SS и выходной сигнал ALE. Этот режим дает возможность останова после выполнения каждой команды. При подаче сигнала (низким уровнем напряжения) микроЭВМ завершает выполнение команды и делает останов; при этом на шинах порта P0 находятся восемь младших, а на шинах порта P2 — четыре старших

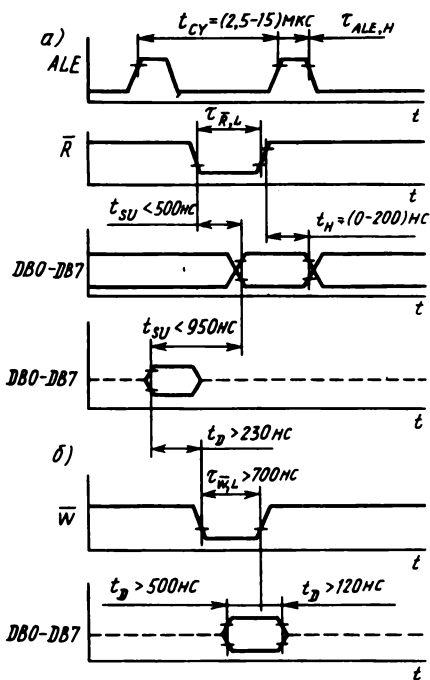


Рис. 4.12. Временные диаграммы обмена данными между микроЭВМ и внешним ОЗУ

разряда адреса следующей команды. Если микроЭВМ находится в состоянии останова, то данные ввода — вывода порта РО и младших разрядах порта Р2 отсутствуют. Эту информацию можно фиксировать внешним устройством по фронту сигнала ALE.

Режим программирования в микросхеме КМ1816ВЕ48 используется для записи информации во внутреннюю память программ.

Микросхемы по входам и выходам совместимы со схемами ТТЛ-серий. Для выводов CR1, CR2, CLR необходимо применять микросхемы ТТЛ-серий с открытым коллектором.

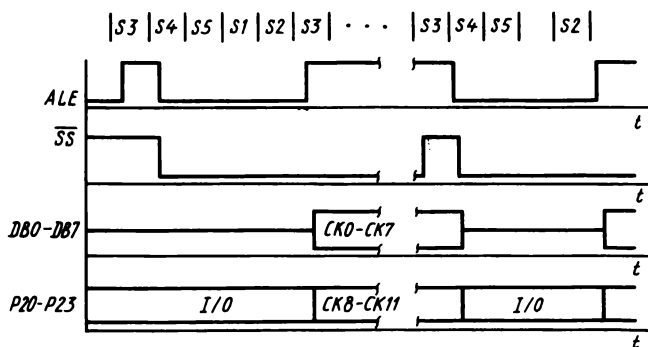


Рис. 4.13. Временные диаграммы режима пошагового выполнения программы

Двунаправленные и квазидвунаправленные выводы рекомендуется подключать через двунаправленные магистральные формирователи серии К589 или аналогичные формирователи.

Структура двунаправленной магистрали DB0—DB7 (порт P0) микроЭВМ позволяет использовать в составе системы интерфейсные микросхемы серии К580 (рис. 4.14).

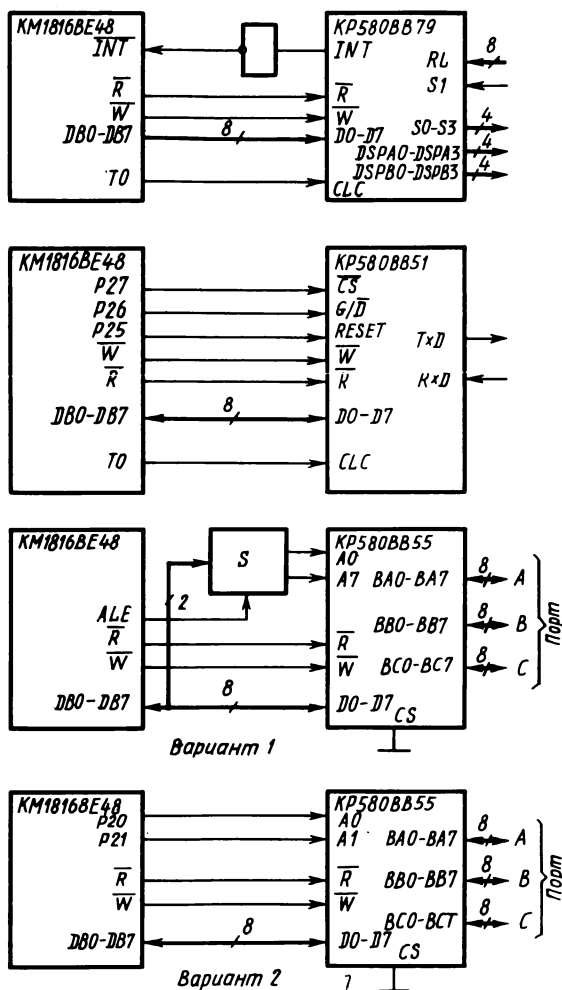


Рис. 4.14. Варианты схем подключения интерфейсных БИС

В системе можно использовать также последовательный интерфейс ввода — вывода КР580ВВ51, параллельный интерфейс ввода — вывода КР580ВВ55, контроллер КР580ВВ79. Обмен информацией в этом случае осуществляется так же, как и при обращении к внешней памяти данных с помощью команд типа MOVX.

4.3. Система команд микроЭВМ

Система команд включает в себя 96 команд, из которых 68 команд — 1-байтовые. В 2-байтовых командах байт 1 несет информацию о коде команды, байт 2 представляет собой непосредственные данные или младшие разряды адреса следующей команды. Большинство команд (53) выполняются за один машинный цикл, 43 команды, в том числе 15 1-байтовых, выполняются за два машин-

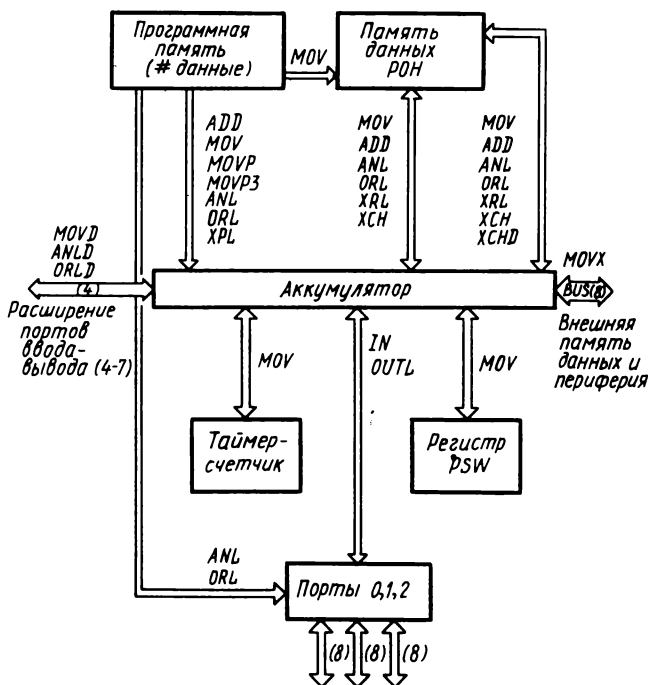


Рис. 4.15. Структура процесса прохождения данных между узлами микроЭВМ

Таблица 4.3

Мнемоника	Код операции, разряды 7, 6, 5, 4, 3, 2, 1, 0	Число циклов	Число байтов команды	Комментарий
Команды передачи данных				
MOV A, R _i (i = 0 - 7)	11111 (i) ₂	1	1	Пересылка содержимого RОН в A
MOV R _i , A (i = 0 - 7)	10101 (i) ₂	1	1	Пересылка содержимого A в RОН
MOV A, @ R _i (i = 0, 1)	1111000 i	1	1	Пересылка косвенно адресуемого байта памяти ЗУ в A
MOV @ R _i , A (i = 0, 1)	1010000 i	1	1	Пересылка содержимого A в косвенно адресуемый байт памяти ЗУ
MOVX A, @ R _i (i = 0, 1)	1000000 i	2	1	Пересылка косвенно адресуемого байта памяти внешнего ЗУ данных в A
MOVX @ R _i , A (i = 0, 1)	1001000 i	2	1	Пересылка содержимого A в косвенно адресуемый байт памяти внешнего ЗУ данных
MOV R _i , # d (i = 0 - 7)	10111 (i) ₂	2	2	Пересылка непосредственных данных в RОН
MOV @ R _i , # d (i = 0 - 7)	1011000 i	2	2	Пересылка непосредственных данных в косвенно адресуемый байт памяти ЗУ
MOV A, # d	0010000011	2	2	Пересылка непосредственных данных в A
MOV A, PSW	11000111	1	1	Пересылка содержимого PSW в A
MOV PSW, A	11010111	1	1	Пересылка содержимого A в PSW
MOV A, T	01000010	1	1	Пересылка содержимого регистра таймера-счетчика в A
MOV T, A	01100010	1	1	Пересылка содержимого A в регистр таймера-счетчика
MOV A, @ A	10100011	2	1	Пересылка в A содержимого косвенно адресуемого байта памяти ЗУ

Мнемоника	Код операции, разряды 7, 6, 5, 4, 3, 2, 1, 0	Число циклов	Число байтов команды	Комментарий
MOV P3 A, @ A	11100011	2	1	много байта памяти ЗУ программ на текущей странице
XCH A, R _i (i = 0 — 7)	00101 (i) ₂	1	1	Пересылка в A косвенно адресуемого байта памяти страницы ЗУ программ
XCH A, @ R _i (i = 0, 1)	0010000 i	1	1	Обмен содержимого A и R _{0H}
XCHD A, @ R _i (i = 0, 1)	0011000 i	1	1	Обмен содержимого A и косвенно адресуемого байта памяти ЗУ
SWAR A	01000111	1	1	Обмен младшего полубайта содержимого A
IN A, P _p (p = 1, 2)	000010 (P) ₂	2	1	Обмен младших полубайтов содержимого A
INS A, BUS	00001000	2	1	Стробируемый ввод данных из порта P1 (P2) в A по окончании сигнала PROG
OUTL P _p , A (p = 1, 2)	001110 (P) ₂	2	1	Стробируемый ввод данных из порта P0 в A по окончании сигнала R
OUTL BUS, A	00000010	2	1	Вывод с фиксацией содержимого A в порт P1 (P2)
MOVD A, P _p	000011 nn	2	1	Вывод содержимого A в порт P0
MOVD P _p , A	001111 nn	2	1	Стробируемый ввод данных из порта 4 (5—7) в младший полубайт A (с обнулением старшего полубайта)
				Стробируемый вывод младшего полубайта A на порт-расширитель 4 (5—7) (с сохранением старшего полубайта)

Команды преобразования данных

ANL A, R _i (i = 0 — 7)	01011 (i) ₂	1	1	Логическое И содержимого A и R _{ON}
ORL A, R _i (i = 0 — 7)	01001 (i) ₂	1	1	Логическое ИЛИ содержимого A и R _{ON}
XRL A, R _i (i = 0 — 7)	11011 (i) ₂	1	1	Исключающее ИЛИ содержимого A и R _{ON}
ANL A, @ R _i (i=0,1)	0101000 i	1	1	Логическое И содержимого A и косвенно адресуемого байта памяти 3У
ORL A, # d	01000011	2	2	Логическое ИЛИ содержимого A с непосредственными данными
XRL A, # d	11010011	2	2	Исключающее ИЛИ содержимого A с непосредственными данными
ANL P _p , # d (p = 1, 2)	10010 (P) ₂	2	2	Логическое И фиксированных данных порта P1 (P ₂) с непосредственными данными
ANL BUS, # d	10011000	2	2	Логическое И фиксированных данных порта P0 с непосредственными данными
ORL P _p , # d	100010 (P) ₂	2	2	Логическое ИЛИ фиксированных данных порта P1 (P ₂) с непосредственными данными
ORL BUS, # d	10001000	2	2	Логическое ИЛИ фиксированных данных порта P0 с непосредственными данными
ANLD P _p , A	100111 nn	2	1	Стробируемый вывод младшего полубайта A на порт-расширитель 4 (5—7) (с сохранением содержимого A) по окончании сигнала через разряды 3—0 порта P2. По началу сигнала PROG выдача через разряды 3—0 порта P2 кода-признака логической операции И и адреса порта-расширителя 4 (5—7)
ORLD P _p , A	100011 nn	2	1	Стробируемый вывод младшего полубайта A на порт-расширитель 4 (5—7) (с сохранением содержимого A) по окончании сигнала PROG через разряды 3—9 порта P2. По началу сигнала PROG выдача через разряды 3—0 порта P2 кода-при-

Мнемоника	Код операции, разряды 7, 6, 5, 4, 3, 2, 1, 0	Число циклов	Число байтов команды	Комментарий
RRA	01110111	1	1	знака логической операции ИЛИ и адреса порта- расширителя 4 (5—7)
RLA	11100111	1	1	Циклический сдвиг содержимого A вправо на 1 разряд без переноса
RRC A	01100111	1	1	Циклический сдвиг содержимого A влево на 1 раз- ряд без переноса
RLC A	11110111	1	1	Циклический сдвиг содержимого A вправо на 1 разряд с переносом
INC A	00010111	1	1	Циклический сдвиг содержимого A влево на 1 раз- ряд без переноса
INC R _i (i = 0 — 7)	00011 (i) ₂	1	1	Инкрементирование содержимого A
INC @ R _i (i = 0, 1)	0001000 i	1	1	Инкрементирование содержимого RОН
DEC A	00000111	1	1	Инкрементирование косвенно адресуемого байта памяти 3У
DEC R _i (i = 0 — 7)	11001 (i) ₂	1	1	Декрементирование содержимого A
DJNZ R _i , ADDRESS (i = 0 — 7)	11101 (i) ₂	2	2	Декрементирование содержимого RОН и тести- рование его на ноль
ADD A, R _i (i = 0 — 7)	01101 (i) ₂	1	1	Сложение содержимого RОН и содержимого A
ADDC A, R _i (i = 0 — 7)	01111 (i) ₂	1	1	Сложение содержимого RОН, содержимого A и разряда переноса
ADD A, @ R _i (i = 0, 1)	0110000 i	1	1	Сложение косвенно адресуемого байта памяти 3У с содержимым A

ADDC A, Δ R _i (i = 0, 1)	01110001	1	1	Сложение содержимого A, разряда переноса и косвенно адресуемого байта памяти ZU
ADD A, # d	00000011	2	2	Сложение непосредственных данных с содержимым A
ADDC A, # d	00010011	2	2	Сложение разряда переноса и непосредственных данных с содержимым A
CLR A	00100111	1	1	Обнуление A
CPL A	00110111	1	1	Инвертирование содержимого A
DA A	01010111	1	1	Десятичная коррекция
CLR C	10010111	1	1	Обнуление разряда переноса
CPL C	10100111	1	1	Инвертирование разряда переноса
CLR F0	10000101	1	1	Обнуление флагового разряда F0
CLR F1	10100101	1	1	Обнуление флагового разряда F1
CPL F0	10010101	1	1	Инвертирование содержимого флагового разряда F0
CPL F1	10110101	1	1	Инвертирование содержимого флагового разряда F1
EN I	00000101	1	1	Разрешение внешних прерываний
DIS I	00010101	1	1	Запрещение внешних прерываний
EN TCNTI	00100101	1	1	Разрешение прерываний по переполнению таймера-счетчика
DIS TCNTI	00110101	1	1	Запрещение прерываний по переполнению таймера-счетчика
STRT T	01010101	1	1	Запуск таймера
STRT CNT	01000101	1	1	Запуск счетчика отрицательных перепадов на выводе T1
STOP TCNT	01100101	1	1	Останов таймера-счетчика
ENTO CLK	01110101	1	1	Разрешение вывода импульсов синхронизации

Мнемоника	Код операции, разряды 7, 6, 5, 4, 3, 2, 1, 0	Число циклов	Число байтов команды	Комментарий
SEL MB 0	11100101	1	1	Выбор нулевого банка памяти программ
SEL MB 1	11110101	1	1	Выбор первого банка памяти программ
SEL RB 0	11000101	1	1	Выбор нулевого банка рабочих регистров памяти данных
SEL RB 1	11010101	1	1	Выбор первого банка рабочих регистров памяти данных
NOP	00000000	1	1	Нет операции

Команды передачи управления

	A ₁₀ , A ₉ , A ₈ 10100	2	2	Вызов подпрограммы
CALL ADDRESS				
RET	10000011	2	1	Возврат из подпрограммы обслуживания прерываний
RETR	10010011	2	1	Возврат из подпрограммы обслуживания прерываний с восстановлением содержимого PSW
JMP ADDRESS	A ₁₀ , A ₉ , A ₈ 00100	2	2	Безусловный переход по адресу, указанному в байте 2 команды
JMPP @ A	10110011	2	2	Безусловный переход по адресу, содержащемуся в A, внутри текущей страницы памяти программы
JC ADDRESS	11110110	2	2	Переход, если разряд переноса установлен в 1
JNC ADDRESS	11100110	2	2	Переход, если разряд переноса установлен в 0
JZ ADDRESS	11000110	2	2	Переход, если содержимое A равно нулю

JNZ ADDRESS	10010110	2	2	Переход, если содержимое A не равно нулю
JFO ADDRESS	10110110	2	2	Переход, если флаговый разряд F 0 установлен в 1
JF1 ADDRESS	01110110	2	2	Переход, если флаговый разряд F1 установлен в 1
JTO ADDRESS	00110110	2	2	Переход, если уровень сигнала на входе T0 высокий
JNT0 ADDRESS	00100110	2	2	Переход, если уровень сигнала на входе T0 низкий
JT1 ADDRESS	01010110	2	2	Переход, если уровень сигнала на входе T1 высокий
JNT1 ADDRESS	01000110	2	2	Переход, если уровень сигнала на входе T1 низкий
JTF ADDRESS	00101110	2	2	Переход, если флаговый разряд таймера установлен в 1
JNI ADDRESS	10000110	2	2	Переход, если на входе прерывания низкий уровень
JB _b ADDRESS (b=0—7)	(b) ₂ 10010	2	2	Переход, если указанный в коде операции разряд A установлен в 1

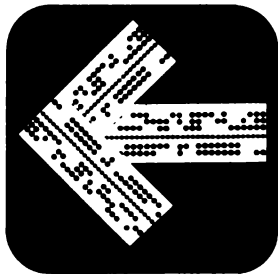
Примечания: 1. Косвенная адресация байта внутренней памяти 3У данных выполняется по адресу в РОН (разряды 0—5).
 2. Фиксация содержимого A при выводе в порт P0 возможна только при работе с внутренней памятью 3У программ. 3. При выполнении команд MOVD A, P_p; MOVD P_p, A; ANLD P_p, A; ORLD P_p, A по началу сигнала PROG на разряды 2—3 порта P2 выводится соответствующий команде код 00, 10, 11 или 01, а на разряды 0—1 порта P2 — код номера порта-расширителя pp, уменьшенный на 4.

ных цикла. Выполнение 2-байтовых команд за два цикла связано с дополнительным обращением к программной памяти. В основном 1-байтовые команды выполняются за один цикл, однако команды, требующие обращения к внешним устройствам, а также команды RET и RETR выполняются за два цикла.

В структуре системы команд микроЭВМ различают команды передачи данных, преобразования данных и передачи управления (табл. 4.3).

Прохождение данных между узлами микроЭВМ при выполнении команд показано на рис. 4.15.

Программное обеспечение комплекса технических средств микроЭВМ «Электроника К1-30»



Состав и характеристики программного обеспечения комплекса технических средств (КТС) «Электроника К1-30» («Электроника МС 0401») обусловлены основным назначением КТС — разработка и отладка программного обеспечения микроЭВМ и контроллеров на основе микропроцессорных комплектов серий К580 и К1810 и однокристалльных микроЭВМ серии К1816.

В связи с этим дисковая операционная система (ДОС) не содержит средств поддержки многопользовательного режима работы, режима работы в реальном времени, но обеспечивает возможность работы с обширной номенклатурой внешних устройств, имеет гибкую систему работы с файлами, допускает расширение за счет включения дополнительных модулей, разработанных пользователем, и использование средств резидентной части ДОС пользователем при разработке программ.

В составе ПО КТС «Электроника К1-30» также нет языков программирования, ориентированных на решения прикладных задач, таких, как ФОРТРАН, БЭЙСИК, ПАСКАЛЬ, но имеются ассемблеры и кросс-ассемблеры для микропроцессоров и однокристалльных микроЭВМ, интерпретаторы и отладчики, необходимые для разработки и отладки системного и прикладного ПО микроЭВМ и контроллеров на базе микропроцессорных комплектов К580, К1810 и однокристалльных микроЭВМ серии 1816.

Программное обеспечение КТС постоянно расширяется за счет новых кросс-систем программирования, ориентированных на разработку и отладку программного обеспечения вновь разрабатываемых микропроцессорных комплектов и однокристалльных микроЭВМ.

5.1. Дисковая операционная система

Дисковая операционная система представляет собой диалоговую систему, рассчитанную на работу с одним пользователем. Она обеспечивает удобный диалог пользователя с системой через дисплей, файловую организацию программ и данных, работу с файлами, распространяет файловую организацию на взаимодействие системы с устройствами ввода — вывода, обеспечивает доступ к возможностям операционной системы из программ пользователя через набор системных вызовов, предоставляет

возможности как диалогового, так и пакетного запуска на выполнение программ.

Имеются ряд трансляторов с языков программирования, загрузчики, редактор связей, библиотечарь и большое число сервисных программ, работающих в рамках ДОС. Другие трансляторы, сервисные и прочие программы могут включаться пользователем в конфигурацию ДОС.

Операционная система состоит из трех программ, содержащихся на системном диске в виде отдельных файлов: начальный загрузчик, резидент, интерпретатор команд.

При пуске ДОС вначале с помощью команд операционной системы, содержащейся в ПЗУ КТС «Электроника К1-30», с диска загружается начальный загрузчик, в свою очередь, начальный загрузчик загружает резидент системы, формирует структуры, зависящие от конфигурации текущей системы ввода — вывода, отключает ПЗУ и на его адресное пространство подключает ОЗУ, загружает с диска программу-монитор объемом 2 К байт, а затем выводит на системный пульт заголовок системы и передает управление интерпретатору команд ДОС. После того как интерпретатор команд выведет на пульт побуждающий символ «<>», пользователь может обращаться к системе со своими заданиями.

Резидент ДОС содержит реализуемые в форме подпрограмм наиболее часто используемые обрабатывающими программами средства работы с файлами и с системой в целом. Обращения к этим подпрограммам называются системными вызовами. Программные модули, обеспечивающие системные вызовы, включены в системную библиотеку ДОС, поэтому они доступны также и прикладным программам. Тем самым пользователю предоставляется возможность эффективного использования средств резидентной части ДОС.

Для выполнения заданий пользователь должен ввести с пульта имя файла, содержащего программу, которая реализует данное задание. Таким файлом может быть как системный файл стандартной обслуживающей программы, так и несистемный файл, содержащий программу пользователя.

Строка с именем файла вводится после побуждающего символа интерпретатора команд. В этой же строке могут указываться параметры, требуемые для обработки задания. Символьная строка, вводимая с пульта после побуждающего символа, называется командной строкой.

Она может вводиться не только с пульта, но и с дискового файла и любого другого устройства ввода. Такой режим обработки заданий называется пакетным.

Системный интерпретатор команд загружает программу, содержащуюся в файле, указанном в командной строке, и передает ей управление. Командная строка, состоящая из имени файла и параметров задания, доступна для загруженной программы. Если имя файла командной строки совпадает с именем одной из встроенных операций интерпретатора команд, то загрузка не производится и интерпретатор выполняет соответствующую операцию. Встроенными операциями интерпретатора команд являются:

DEBUG — перевод системы в режим отладки программы;

JOB — формирование командного файла и выполнение пакета;

END JOB — отмена пакетного выполнения заданий;

OUTPUT — перевод протокола работы с ДОС на устройство вывода;

/ — пакетное выполнение заданий;

@ — управляемый вывод текстовых файлов;

? — справочная операция.

Обслуживающие программы ДОС и программы пользователя обрабатываются интерпретатором команд одинаково. Таким образом, вводя новые программные файлы, пользователь может расширять возможность системы.

Интерпретатор команд вначале получает управление от начального загрузчика. После выполнения любой из обслуживающих программ ДОС управление возвращается интерпретатору команд. Этим достигается интерактивность процесса обработки заданий. Если текущее задание реализуется программой пользователя, то для нормального продолжения процесса обработки эта программа после завершения должна возвращать управление интерпретатору команд с помощью системного вызова EXIT.

Способ организации файлов, принятый в ДОС, позволяет наиболее полно использовать возможность гибкого диска как устройства с непосредственным доступом. Единицей информации при размещении файла на диске является блок, состоящий из 128 байт и имеющий емкость одного сектора диска. Каждый файл зависит от длины и состоит из определенного числа информационных и указательных блоков. Информационные

блоки содержат собственно данные файла, а указательные блоки отмечают, какие информационные блоки и в какой последовательности составляют файл. На каждые 62 информационных блока файла выделяется один указательный блок.

Позиция информационного блока отмечается в указательном блоке парой байтов, задающих физические номера дорожки (0—76) и сектора (1—26). Первые четыре байта указательного блока отмечают позиции предыдущего и последующего указательных блоков. Таким образом, для каждого файла строится цепочка указательных блоков, последний из которых может быть неполным.

Однако детали физического размещения файла на носителе пользователю нет необходимости знать. При выполнении любых действий с файлами (открытие файла, чтение файла, запись в файл, перемещение маркера файла, закрытие файла) поддержание структуры файлов и обеспечение требуемых дисковых операций осуществляет система.

Принятое в ДОС обращение к устройствам и файлам по именам существенно облегчает работу с ними. Система содержит список адресов устройств, заложенный при ее генерации, и список адресов файлов, модифицируемый при создании или удалении файла. Имена файлов преобразуются в адреса в результате выполнения процедуры открытия файла. При этом информация о файле заносится в системные таблицы и, если необходимо, отводятся буферы ввода — вывода. Когда файл закрывается, информация о нем удаляется из системных таблиц, буферы освобождаются и в случае создания или изменения дискового файла формируется новая или модифицируется имеющаяся строка в оглавлении диска.

При работе с обслуживающими программами файлы по именам идентифицирует только пользователь. Сама же система различает файлы не по именам, а по номерам, которые присваиваются файлам при их открытии. Этим достигается более наглядный доступ к файлам для пользователя и более простой доступ для операционной системы. После закрытия файла его номер освобождается и используется для идентификации следующего файла. Одновременно может быть открыто до шести файлов.

Перед выполнением обмена в программе пользователя необходимо открыть файл с помощью системного вызова OPEN. Системные вызовы чтения файла READ,

записи в файл WRITE, перемещения маркера файла SEEK и закрытия файла CLOSE оперируют не с именем файла, а с его номером в системе. Однако в отличие от работы с обслуживающими программами при написании программы это не является неудобством, так как большинство языков программирования позволяет вводить для номеров файлов идентификаторы не менее информативные, чем имена файлов.

Системные диски — это диски, содержащие ДОС. Поставляемые системные диски помимо ДОС содержат файлы обслуживающих программ и трансляторов. Несистемные диски не содержат ДОС. Как правило, они являются рабочими, т. е. дисками, содержащими прикладные программы на всех или отдельных стадиях разработки.

Для того чтобы диск использовать в ДОС, его необходимо предварительно инициализировать с помощью обслуживающей программы IDISK или FORMAT. При этом для несистемного диска записываются четыре, а для системного диска — семь служебных файлов. Общими служебными файлами для системных и несистемных файлов являются:

KRIS. DIR — оглавление диска;

KRIS. MAP — карта занятых секторов;

KRIS. LAB — метка диска;

KRIS. TO — начальный загрузчик.

О г л а в л е н и е д и с к а представляет собой каталог файлов, имеющихся на диске. Каталог состоит из 200 описаний файлов, каждое длиной 16 байт. Описание файла включает в себя признак использования (описание имеющегося файла, описание удаленного файла, описание свободного), имя файла, расширение имени файла, атрибуты файла, число информационных блоков, длину последнего информационного блока и адрес первого указательного блока файла.

Атрибутами файла являются атрибуты невидимости, защиты от записи, форматные, системные. Атрибуты файла можно устанавливать и отменять с пульта с помощью обслуживающей программы ATTRIB и программно посредством системного вызова ATTRIB. Файлы с атрибутом невидимости пропускаются при выводе оглавления диска с помощью обслуживающей программы DIR. Атрибут защиты от записи позволяет избежать случайного удаления или изменения файла. Форматный атрибут присваивается только служебным файлам, которые называются также форматными, так как они определяют

форматы информационных структур диска. В отличие от других файлов служебные файлы занимают фиксированные адреса на диске и не могут быть скопированы посредством системной обслуживающей программы COPY. Форматный атрибут обеспечивает также защиту служебного файла от непреднамеренного удаления или изменения. Системный атрибут отмечает файлы системных обслуживающих программ и используется при создании системных дисков обслуживающими программами FORMAT и COPY.

Карта занятых секторов используется для записи новых файлов на свободные адреса диска. Один бит карты занятых секторов отмечает, занят или свободен блок, адрес которого соответствует позиции данного бита. При записи, удалении и изменении длины каждого файла система корректирует содержимое карты занятых секторов.

Метка диска является именем диска и служит для его идентификации. Формат метки диска совпадает с форматом имен файлов ДОС, т. е. метка диска состоит из имени длиной 1—6 символов и необязательного расширения длиной 1—3 символов, следующего после точки «.». Метка присваивается диску при его инициализации и используется при выводе оглавления диска, а также для контроля правильности смены диска в дисковом устройстве всеми обслуживающими программами, которые предусматривают такую смену.

Начальный загрузчик системных дисков предназначен для подготовки ДОС к работе и загрузке резидента ДОС, монитора и интерпретатора команд. Начальный загрузчик несистемных дисков содержит программу вывода на пульт сообщения

НЕСИСТЕМНЫЙ ДИСК, ПРОБУЙТЕ ДРУГОЙ,
которое выдается при попытке загрузить ДОС с несистемного диска.

Дополнительными служебными файлами системных дисков являются:

KRIS.BIN — резидент ДОС;

KRIS.CLI — интерпретатор команд;

KRIS.MON — монитор.

Резидент ДОС занимает 12 К байт ОЗУ начиная с младших адресов. Он реализует системные вызовы, доступные системному интерпретатору команд, нерезидентным обслуживающим программам и программам пользователя. К резидентной области ДОС принадлежит

также область буферов, начинающаяся с адреса 3000H. Длина области буферов зависит от числа одновременно открытых дисковых файлов в выполняемом задании и изменяется от 180H до 680H. Область буферов в файле резидента ДОС не резервируется.

Интерпретатор команд предназначен для обработки командной строки, загрузки и запуска программ и поддержания диалога с пользователем.

Монитор управляет выполнением процедур ввода — вывода и реализует элементарные отладочные функции.

Как системные, так и несистемные диски могут иметь дополнительный служебный файл KRIS.BAD. Этот файл формируется на диске при его инициализации в случае обнаружения секторов диска, операции обмена с которыми происходят с ошибкой. Файл KRIS.BAD собирается из неработоспособных блоков, при этом из обращения исключаются соответствующие секторы диска. В тех случаях, когда адрес неработоспособного блока попадает в область адресов, выделенную для служебных файлов, инициализация диска прекращается.

Все обслуживающие программы содержатся в одноименных файлах абсолютных объектных модулей. ДОС поставляется в комплекте со следующими программами:

ATTRIB — изменение атрибутов одного или группы дисковых файлов;

BINOBJ — преобразование абсолютного модуля из двоичного формата в формат ДОС. Абсолютный двоичный формат используется другими операционными системами и имеет более прочную структуру, состоящую из набора секций, которые содержат начальный адрес, длину секции и непрерывные по адресам данные;

COPY — передача содержимого исходного файла или группы исходных файлов в новый файл или группу новых файлов. Данная обслуживающая программа применяется также для объединения нескольких файлов в один. Входным файлом может служить дисковый файл или любое устройство ввода, а выходным — дисковый файл или любое устройство вывода. Широкий набор управляющих параметров в сочетании с групповыми именами файлов делает данную обслуживающую программу удобным средством создания новых файлов;

DELETE — удаление одного файла, нескольких файлов или группы файлов;

DIR — вывод оглавления диска;

FORMAT — инициализация диска. При инициализации системного диска с исходного диска копируются все системные файлы;

HEXOBJ — преобразование абсолютного модуля из шестнадцатеричного формата в формат ДОС;

IDISK — инициализация диска. Для системных и несистемных дисков формируются только форматные файлы;

OBJHEX — преобразование абсолютного модуля из формата ДОС в шестнадцатеричный формат;

RENAME — переименование дискового файла;

SUBMIT — пакетное выполнение программ. Перед использованием данной программы пользователь должен создать файл описания последовательности команд, в котором допускается введение до 10 формальных параметров. В обращении к программе **SUBMIT** указываются имя файла описания последовательности команд и фактические параметры, которые заменяют формальные параметры в создаваемом программой **SUBMIT** командном файле. Во время пакетного выполнения программ возможен переход на ввод заданий из исходного пульта и обратно;

MOSDOS — формирование файлов ДОС из файлов операционной системы МОС2 и файлов МОС2 из файлов ДОС;

VERIFY — контроль информации, содержащейся на системном диске.

В комплект поставки программного обеспечения КТС «Электроника МС0401» помимо ДОС и файлов обслуживающих программ входят файлы следующих программ системы программирования:

ASM80 — транслятор языка ассемблера микропроцессорного набора серии 580;

ASXREF — программа формирования перекрестных ссылок в листинге ассемблерного модуля;

EDIT — командный редактор текстов;

CREDIT — экранный редактор текстов;

LIB — библиотечка объектных модулей;

LINK — редактор связей (корневая фаза — **LINK** оверлей, т. е. **LINK.OVL**);

LOCATE — абсолютный настройщик.

Резидент ДОС реализует следующие системные вызовы:

OPEN — открытие файла для операций ввода — вывода;

READ — передача данных из файла в память;

WRITE — передача данных из памяти в файл;
SEEK — перемещение указателя дискового файла;
RESCAN — перемещение указателя строчно-редактируемого файла к началу строки;
CLOSE — закрытие файла;
SPATH — получение информации о файле;
DELETE — удаление дискового файла;
RENAME — переименование дискового файла;
ATTRIB — изменение атрибутов дискового файла;
CONSOL — переназначение пультовых устройств;
WHOCON — определение текущего назначения системному пульту;

ERROR — вывод сообщения об ошибке на системный пульт;

LOAD — загрузка программы и передача управления;
EXIT — завершение программы и возврат в ДОС;
DISK — непосредственное выполнение дисковой операции;

FATSW — управление возвратом при фатальной ошибке.

Монитор реализует следующие системные вызовы:

CI — ввод символа с системного пульта;

CO — вывод символа на системный пульт;

CIO — ввод символа с системного пульта с отражением;

RI — ввод байта с системного устройства ввода;

RO — вывод байта на системное устройство вывода;

LO — вывод символа на системное устройство печати;

CSTS — определение состояния ввода с системного пульта;

IODEF — расширение системы ввода — вывода;

IOCHK — проверка конфигурации системы ввода — вывода;

IOSET — установка конфигурации системы ввода — вывода;

MEMCK — определение максимального доступного адреса оперативной памяти для программы пользователя;

DIAGNS — вывод диагностических сообщений.

При сборке объектных модулей в командной строке редактора связей LINK необходимо указать имя файла системной библиотеки **SISTEM.LIB**, которая содержит модули обработки системных вызовов ДОС.

Отладчик **UNIDEB** предназначен для повышения производительности отладки программ, написанных на языках высокого уровня (PL/M-80, ФОРТРАН-80,

ПАСКАЛЬ и др.). Принцип действия отладчика UNIDEB основан на исследовании отладочной информации, содержащейся в файле собранного объектного модуля. Для каждого модуля, который транслировался с управляющим параметром DEBUG, созданы таблицы адресов общих имен, локальных имен и номеров выполняемых операторов. Доступ к этим таблицам позволяет охватить отлаживаемую программу управлением по событиям. Такими событиями являются достижение оператора с заданным номером, выполнение определенного оператора заданное число раз и изменение значения указываемой переменной или элемента массива (любое значение или его изменение в заданных пределах). События, задаваемые пользователем, должны соответствовать узловым моментам текущего этапа отладки программы. При выявлении требуемого события в отлаживаемой программе ее выполнение прерывается и пользователь получает доступ к интересующим его элементам программы. По желанию при выявлении требуемого события могут выводиться интересующие пользователя значения всех объектов или объектов, заданных в списке, без останова выполнения отлаживаемой программы.

Интерактивные прерывания по событиям повышают гибкость отладки, так как после исследования интересующих элементов уточняется направление поиска ошибки. При традиционных способах отладки не всегда можно предсказать, значения каких элементов следует вывести и в каком направлении пойдет выполнение программы.

Описанный механизм отладки позволяет обнаружить несколько ошибок за один прогон отлаживаемой программы. При выявлении очередной ошибки можно задать правильные значения соответствующих переменных и массивов и продолжить отладку.

Преимуществом использования отладчика UNIDEB (по сравнению с включением в отлаживаемую программу средств отладочной печати) является то, что отлаживаемая программа не содержит никаких специальных средств, необходимых для его использования и отладчик, в свою очередь, не вносит изменений в объектный модуль отлаживаемой программы. Поэтому один и тот же отлаживаемый модуль может выполняться и с помощью отладчика, и автономно.

Отладчик UNIDEB выполняется, находясь в памяти одновременно с отлаживаемой программой. Особен-

ностью его реализации является то, что при каждом пуске он самостоятельно настраивается на адреса, свободные от сегментов отлаживаемой программы.

Отладчик UNIDEB включает в себя файлы с именами UNIDEB, UNIDEB.OVL, UNIDEB.OV0, UNIDEB.OV1 и UNIDEB.HLP. Файл UNIDEB содержит начальную фазу отладчика, файл UNIDEB.OVL — собранный объектный модуль для основной фазы, которая находится в памяти одновременно с отлаживаемой программой. При пуске отладчика начальная фаза настраивает сегменты основной фазы на свободные адреса. Настройка производится на основании содержащейся в файле абсолютного модуля отлаживаемой программы информации о сегментах этого модуля.

Файлы UNIDEB.OV0 и UNIDEB.OV1 предназначены для обработки объектов программы соответственно типов INTEGER и REAL. Если отлаживаемые модули написаны на языке программирования PL/M-80, не обеспечивающем сложных типов данных, то файлы UNIDEB.OV0 и UNIDEB.OV1 следует временно переименовать или изъять из диска, на котором содержатся остальные файлы отладчика. Этим достигается, во-первых, убыстрение настройки основной фазы отладчика, а во-вторых, выделение большего объема памяти под таблицы описания отладочной информации, в результате чего уменьшается число обращений к диску.

Файл UNIDEB.HLP содержит краткое описание отладчика и форматов его команд.

Отладчик UNIDEB реализован в двух вариантах. Первый вариант (версия 3.0) предназначен для отладки программ, объем которых превышает 20 К байт, второй вариант (версия 3.1) используется для отладки модулей и программ объемом до 20 К байт. Первый вариант реализует только основные отладочные функции, обеспечиваемые командами B, C, E, G, H, M, S, V, X и ?. Система команд второго варианта отладчика дополнена командами A, D, F, I, N, Q, T и U, реализующими функции отладки по событиям «изменение данной переменной» и «изменение данной переменной в заданных пределах», трассировку выполнения программ и ряд сервисных функций. Второй вариант отладчика также предоставляет возможность обработки данных фортранных типов INTEGER и REAL.

Назначение команд отладки по событиям следующее:

A — задание и отмена режима отладки по событиям

«изменение данной переменной» и «изменение данной переменной в заданных пределах».

В — задание и отмена режима отладки по событиям «достижение оператора с заданным номером».

С — задание и отмена режима отладки по событию «выполнение данного оператора заданное число раз».

D — удаление переменных и сечений массивов из списка исследуемых переменных.

Е — прекращение работы с отладчиком.

F — заполнение константой вырезки массива.

G — пуск отлаживаемой программы, которая может запускаться со своей точки входа, с точки последнего прерывания и любого оператора отлаживаемого модуля. В команде G можно указывать точки разрыва в дополнение к заданным командой В. При интерактивном прерывании по любому из событий, заданных командами А, В, С или G, все точки разрыва, заданные в команде G, аннулируются.

H — вывод «подсказки», содержащейся в файле UNI-DEV.HLP.

I — дополнение списка исследуемых переменных новыми переменными и сечениями массивов.

M — задание имени отлаживаемого модуля.

N — заполнение холостой командой процессора участка отлаживаемой программы.

Q — вывод данных об отлаживаемой программе — распределении памяти и составе модулей, транслировавшихся с управляющим переключателем DEBUG.

S — исследование и изменение значений базированных переменных и элементов базированных массивов.

T — задание предустановленного способа реакции на события, определяемые командами А, В и С.

U — отмена предустановленного способа реакции на события, определяемые командами А, В и С, и задания интерактивного способа реакции на эти события.

V — задание списка переменных и сечений массивов для последующего исследования с помощью команды X и ограничения области действия предустановленного способа реакции на события, определяемые А, В, и С.

X — диалоговое и неинтерактивное исследование значений общих и локальных переменных, массивов и структур отлаживаемого модуля.

? — вывод сведений о заданных на текущий момент условиях отладки, а также адреса переменной или оператора отлаживаемого модуля.

5.2. Редакторы текстов

Редакторы текстов предназначены для создания и редактирования текстовых данных (в частном случае — исходных программ). В составе программного обеспечения КТС имеется два редактора текстов: командный EDIT и экранный CREDIT. Командный редактор текстов позволяет создавать и редактировать текстовые данные, используя различающиеся по принципу действия терминалы (дисплеи, электрифицированные пишущие машины и др.), экранный — только посредством дисплеев.

По средствам редактирования командный редактор текстов является подмножеством экранного редактора (см. табл. 5.1). Производительность редактирования данных с помощью экранного редактора текстов в несколько раз выше, чем с помощью командного.

Таблица 5.1

Средства редактирования	Редактор	
	CREDIT	EDIT
Средства управления указателем		
Перемещение указателя в начало текста	+	+
Перемещение указателя в конец текста	+	+
Посимвольное перемещение указателя	+	+
Построчное перемещение указателя	+	+
Средства установки, удаления и работы с метками	+	—
Средства редактирования		
Отображение макета	+	+
Вставка текста	+	+
Удаление строк	+	+
Удаление символов	+	+
Пересылка фрагмента текста	+	—
Копирование фрагмента текста	+	—
Поиск последовательности символов	+	+
Поиск и замена последовательности символов	+	+
Поиск и замена последовательности символов с разрешения пользователя	+	—
Возможность использования групповых символов в командах поиска	+	—
Возможность определения области поиска	+	—

Средства редактирования	Редактор	
	CREDIT	EDIT
Средства управления файлами текстовых данных		
Автоматические средства считывания и записи редактируемого текста	+	—
Командные средства считывания и записи редактируемого текста	—	+
Средства считывания и записи нескольких файлов текстовых данных	+	—
Средства объединения команд в строку команд	+	+
Средства циклического выполнения строки команд	+	+
Средства условного и безусловного выхода из цикла	+	—
Средства условного выполнения команд строки	+	—
Средства определения вызова, удаления и отображения макрокоманд	+	—
Средства экранного редактирования		
Вставка последовательности символов	+	—
Замена последовательности символов	+	—
Вставка одного символа	+	—
Удаление последовательности символов	+	—
Удаление одного символа	+	—
Перемещение указателя по экрану	+	—
Управление содержимым экрана	+	—
Вызов макрокоманд, определенных в командном режиме	+	—

5.3. Резидентная система программирования

Резидентная система состоит из ассемблера ASM 80, редактора связей LINK, библиотечка LIB и настройщика адресов LOCATE. Обращение к указанным программам осуществляется посредством интерпретатора команд операционной системы. Типичная последовательность разработки прикладной программы с использованием резидентной системы программирования показана на рис. 5.1.

Подготовка исходных модулей прикладной программы

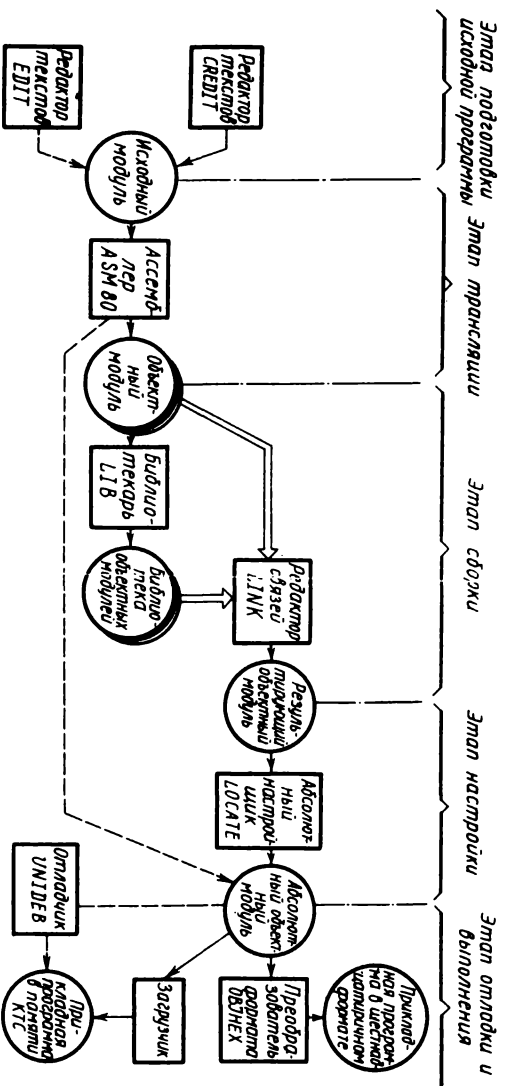


Рис. 5.1. Последовательность разработки прикладной программы на КТС «Электроника К1-30»

на языке EDIT ассемблера осуществляется посредством редактора текстов EDIT или CREDIT.

Каждый исходный модуль транслируется в объектный модуль посредством ассемблера ASM 80, средства которого позволяют создавать абсолютные или переместимые объектные модули.

Переместимый объектный модуль может быть:

- объединен с другими объектными модулями или библиотеками объектных модулей в результирующий объектный модуль посредством редактора связей LINK;

- записан в библиотеку объектных модулей посредством библиотекаря LIB.

Результирующий объектный модуль может быть настроен на абсолютные адреса памяти посредством абсолютного настройщика LOCATE.

Абсолютный объектный модуль может быть:

- загружен в память посредством загрузчика операционной системы для выполнения или отладки (с помощью монитора);

- загружен в память отладчиком UNIDEB с целью проверки правильности функционирования;

- преобразован в абсолютный шестнадцатеричный объектный модуль посредством преобразователя форматов ОВJНЕХ. Шестнадцатиразрядный объектный модуль легко переносится (например, посредством перфоленты) с КТС «Электроника К1-30» на любую другую систему подготовки или обработки данных (например, программатор ППЗУ микросхем).

Язык ассемблера ASM-80 ориентирован на систему команд микропроцессора типа КР580ВМ80А. Он позволяет использовать все функциональные возможности КТС «Электроника МС0401» с той же полнотой, что и при программировании на машинном языке, причем, используя язык ассемблера, можно существенно сократить сроки разработки, повысить качество и надежность прикладных программ.

Алфавит языка ассемблера соответствует графическим символам алфавитно-цифровых наборов 0, 2 или 3, определенных по ГОСТ 19701—74.

Коды машинных команд и регистров микропроцессора представляют символическими именами; адреса команд и данных — в числовой и символической формах.

Для определения непосредственных данных, адресов

памяти и переменных этапа трансляции используются выражения языка. В качестве термов в выражении могут использоваться:

- двоичные, восьмеричные, десятичные, шестнадцатеричные и символьные (литерные) константы;
- символические адреса памяти (имена меток);
- имена переменных этапа трансляции;
- обозначение счетчика адреса ассемблера.

Над термами в выражении выполняются арифметические операции (сложения, вычитания, умножения, деления, вычисления остатка от деления), операции сдвига влево и вправо, поразрядные логические операции (сложения, умножения, сложения по модулю 2, инверсии), операции отношения (равно, не равно, меньше, меньше или равно, больше, больше или равно), операции выделения старшего или младшего байтов.

Значения термов, результатов вычисления операций и выражений определяются ассемблером по модулю 65 536.

Предложения языка ассемблера записываются в свободном формате, т. е. до и (или) после разделителя в любой конструкции языка могут быть вставлены пробелы. Формат предложения состоит из четырех полей, следующих в порядке перечислений: метки, мнемокода, аргументов и комментария.

Поле метки используется для определения имен символического адреса памяти, переменной этапа трансляции или макрокоманды. Поле мнемокода используется для записи символического имени машинной команды, макрокоманды или директивы языка. При записи машинной команды поле аргументов используется для определения аргументов машинных команд. В качестве аргументов может использоваться имя программно-доступного регистра микропроцессора и выражение языка (в частном случае — константа, имя метки, имя переменной этапа ассемблирования). Поле комментария используется для записи пояснения к программе.

Программный модуль на уровне языка ассемблера определяется как набор абсолютных сегментов, сегментов команд, данных стека и дополнительного сегмента, который начинается директивой объявления имени программного модуля, а заканчивается директивой окончания программного модуля. В директиве окончания для главного модуля может быть указана точка передачи управления после его загрузки в память. Для определе-

ния сегментов используются директивы сегментирования программного модуля.

В языке имеются средства определения 8-битовых, 16-битовых и символьных (литерных) данных, резервирования памяти, определения переменных этапа трансляции, условного ассемблирования предложений исходного модуля, управления счетчиков адреса ассемблера; определения внешних и общих имен для связи объектных модулей, определения и вызова макрокоманд.

Ассемблер ASM 80 выполняет трансляцию исходного модуля в объектный за два просмотра. В процессе первого просмотра выделяются все символические имена, определенные пользователем, а также распределяется память для команд и данных. В процессе второго просмотра генерируется абсолютный или переместимый объектный модуль и формируется листинг программы, содержащий объектный код, соответствующий текст исходного модуля и сообщения об ошибках (в случае их обнаружения).

Ассемблер ASM 80 имеет встроенный текстовый макрогенератор.

В текст исходного модуля возможно включение команд управления содержимым листинга (в частности, команд включения в листинг таблицы символических имен и таблицы перекрестных ссылок), управления форматом листинга, включения отладочной информации в объектный модуль, ассемблирования подчиненного файла в процессе ассемблирования главного файла одного исходного модуля, управления встроенным макрогенератором.

Команда обращения к ассемблеру определяет имена файлов исходного модуля, листинга, объектного модуля. В качестве параметров в команде обращения могут использоваться все команды управления ассемблером.

Редактор связей LINK выполняет сборку входных объектных модулей в один результирующий. Имя результирующего модуля определяется по умолчанию или указывается пользователем. Входные объектные модули могут содержаться в отдельных файлах либо в библиотечных файлах.

В процессе сборки одноименные сегменты из входных объектных модулей соединяются в один результирующий сегмент [ссылки на команды и (или) данные присоединяемых сегментов редактируются относительно начала

результатирующего сегмента], удовлетворяются внешние ссылки между входными модулями.

Один или несколько объектных модулей могут объединяться с библиотеками. В этом случае библиотеки просматриваются в поиске модулей, удовлетворяющих внешние ссылки в объединенных модулях. Найденные модули включаются в результирующий.

Имеется возможность реализовать внешние ссылки без присоединения модулей, которые удовлетворяют эти ссылки. Обычно этот способ сборки объектных модулей применяется для удовлетворения внешних ссылок корневой фазы и оверлеев.

В команде обращения к редактору связей указываются перечень имен файлов объектных модулей и, возможно, имена модулей, имя файла результирующего объектного модуля и управляющие параметры.

Абсолютный настройщик LOCATE преобразует переместимый объектный модуль в абсолютный, который может быть загружен в память для отладки или выполнения. В процессе настройки перемещаемым объектам (командам, данным и ссылкам на них) присваиваются абсолютные значения. На этапе настройки адресов может быть изменен порядок следования сегментов модуля, указаны адреса загрузки сегментов, определена точка входа в программу, присвоено имя абсолютному объектному модулю.

В команде обращения к абсолютному настройщику указываются имя входного результирующего объектного модуля, имя выходного абсолютного объектного модуля и управляющие параметры.

Библиотекарь LIB создает при необходимости библиотечный файл, включает в библиотеку новые объектные модули, удаляет из библиотеки хранящиеся там модули, формирует каталог библиотеки. Выполнение требуемых функций библиотекарем осуществляется посредством управляющих команд.

5.4. Кросс-системы программирования

Кросс-системы программирования позволяют выполнять подготовку и отладку программного обеспечения для разрабатываемых микропроцессоров и однокристальных микроЭВМ на серийно выпускаемых технических средствах. При этом становится возможным снижение трудоемкости разработки программного обеспечения и уско-

рение внедрения разрабатываемых микропроцессоров и микроЭВМ. Для однокристальных микроЭВМ типа КМ1816ВЕ48 создание кросс-систем программирования крайне необходимо, так как невозможно создание технических средств для отладки ПО на основе однокристальных микроЭВМ, обладающих достаточными функциональными возможностями из-за ограниченной системы команд и малых объемов резидентной памяти.

Кросс-система программирования КС 48 обеспечивает подготовку прикладных программ для однокристальных микроЭВМ типа КМ1816ВЕ48. Состав кросс-системы: ассемблер ASM 48; интерпретатор INT 48.

Обращение к программам кросс-системы осуществляется посредством интерпретатора команд (загрузчика) операционной системы.

Типичная последовательность разработки прикладной программы с использованием кросс-системы программирования КС48 показана на рис. 5.2.

Подготовка исходной программы на языке ассемблера осуществляется посредством редакторов текстов EDIT или CREDIT. Исходная программа затем транслируется ассемблером в абсолютную шестнадцатеричную программу. Если в результате трансляции обнаруживаются синтаксические ошибки, то исходная программа редактируется, а затем повторно транслируется.

Отсутствие в ассемблере средств перемещения обусловлено особенностями архитектуры и системы команд семейства однокристальных микроЭВМ типа КМ1816ВЕ48, а именно:

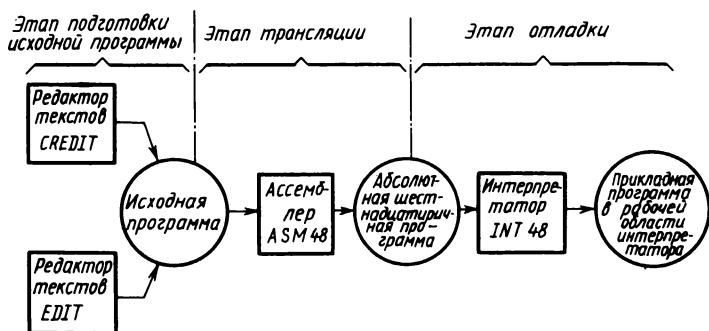


Рис. 5.2. Последовательность разработки прикладной программы с использованием кросс-системы программирования КС48

— небольшим объемом памяти команд (1 К байт резидентной и 3 К байт внешней);

— наличием программно-управляемых банков памяти команд, ограничивающих размер перемещаемой программы до 2 К байт;

— отсутствием условных команд передачи управления между страницами памяти команд, ограничивающих размер перемещаемой программы до 256 байт.

После трансляции с помощью интерпретатора команд INT 48 моделируется выполнение программы. Отложенные программы записываются в ППЗУ однокристалльной микроЭВМ.

Средства ассемблера ASM 48 совместимы с аналогичными средствами ассемблера ASM 80 и обеспечивают создание абсолютных программ. Отличие состоит в том, что файл абсолютной программы формируется в шестнадцатеричном виде. Эта совместимость обеспечивает простоту перехода от программирования на языке ассемблера ASM 80 к программированию на языке ассемблера ASM 48.

Интерпретатор INT 48 позволяет выполнить тестирование, снизить трудоемкость этапов тестирования и отладки, сократить число их повторений и тем самым существенно повысить производительность труда программиста. Для выявления ошибок в прикладной программе предусмотрена возможность распечатки содержимого программно-доступных элементов однокристалльной микроЭВМ (памяти, регистров, портов, специальных триггеров и выводов). Интерпретатор позволяет оценить полноту тестирования программы путем формирования таблицы адресов команд, которые не выполнялись во время интерпретации, а также указать те команды условного перехода, которые выполнялись только при наличии или отсутствии требуемого условия. Если во время тестирования программы были обнаружены ошибки, то интерпретатор позволит существенно облегчить их поиск. Для этого предусмотрены остановки программы при возникновении любого события из множества заданных, выдача строк трассировки, оперативное изменение содержимого регистров и памяти. При обнаружении ошибки в файл исходной программы вносятся исправления, затем процесс тестирования и отладки повторяется.

Кросс-система программирования КС 86 обеспечивает подготовку прикладных программ для контролле-

ров и микроЭВМ на основе микропроцессора типа КМ1810ВМ86.

Состав кросс-системы: ассемблер ASM 86; редактор связей LINK 86; редактор адресов LOC 86; совмещенный редактор связей и адресов QRL 86; библиотечарь LIB 86; конвертор CONV 86; преобразователь форматов OM86.

Типичная последовательность разработки прикладной программы с использованием кросс-системы программирования KC86 показана на рис. 5.3.

Подготовка исходного модуля на языке ассемблера ASM86 осуществляется посредством редакторов текстов EDIT, CREDIT или конвертора CONV86. Исходный модуль затем транслируется ассемблером ASM86 в перемещаемый объектный модуль.

Перемещаемый объектный модуль может быть связан с другими объектными модулями (библиотеками объектных модулей) и преобразован в результирующий объектный модуль посредством редактора связей LINK86; связан с другими объектными модулями (библиотеками объектных модулей) и преобразован в абсолютный объектный модуль посредством совмещенного редактора связей и адресов QRL 86; записан в библиотеку объектных модулей посредством библиотечаря LIB 86.

Результирующий объектный модуль преобразуется в абсолютный объектный модуль посредством абсолютного настройщика LOC 86. Разделение этапа построения прикладной программы на два подэтапа позволяет создавать задачи оверлейной структуры. Абсолютный объектный модуль может быть преобразован в шестнадцатеричный объектный модуль посредством преобразователя форматов ON86.

Все средства ассемблера ASM80 имеются в ассемблере ASM86. Кроме того, ассемблер ASM86 имеет ряд дополнительных возможностей и средств, а именно:

- разделение объектов программы на типы — переменные, метки, константы и др. Проверка соответствия типа ссылки и типа указаний в описании объекта;

- средства определения структур и записей, присущие языкам высокого уровня;

- языковые средства адресации сегментов, команд и данных, присущие микропроцессору типа КМ1810ВМ86;

- оптимизацию объектного кода прикладной программы, допускаемую системой команд микропроцессора типа КМ1810ВМ86;

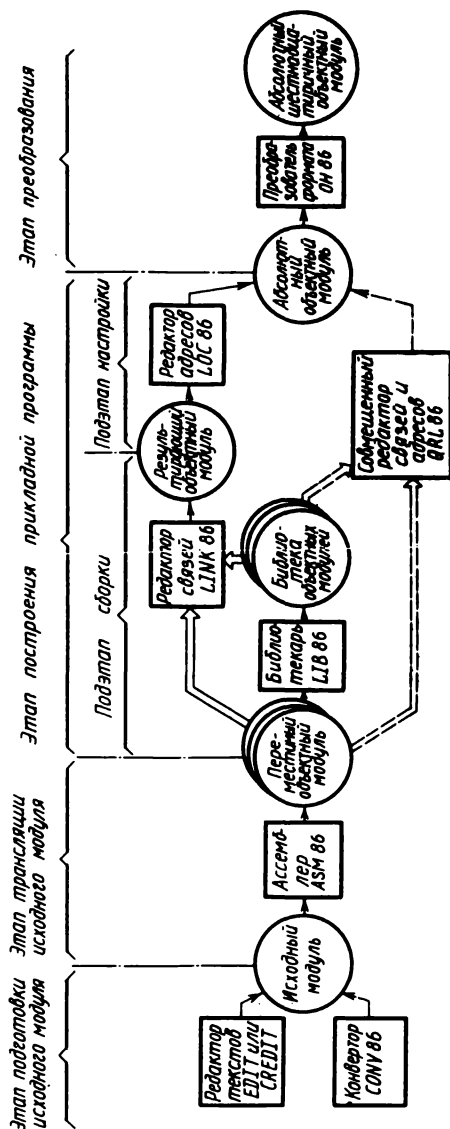


Рис. 5.3. Последовательность разработки прикладной программы с использованием кросс-системы программирования KC86

— встроенный макрогенератор кодов машинных команд.

Все средства программы сборки и настройки LINK, LOCATE и LIB резидентной системы программирования имеются в соответствующих программах сборки и настройки LINK86, LOC86, QRL86 и LIB86 кросс-системы программирования KC86. Кроме того, в программах сборки и настройки кросс-системы программирования имеется ряд дополнительных возможностей, связанных с особенностями программирования микропроцессора типа КМ1810ВМ86. Если прикладная программа имеет простую структуру, то для ее построения вместо программ LINK86 и LOC86 можно использовать совмещенный редактор связей и адресов QRL86.

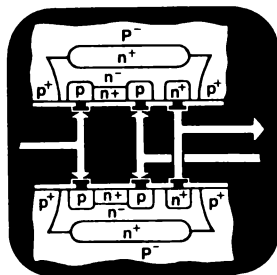
Преобразователь форматов ОН86 позволяет получить прикладную программу в виде абсолютного шестнадцатеричного объектного файла, который может быть перенесен из КТС «Электроника К1-30» на любую другую систему подготовки или обработки данных (например, на программатор ППЗУ микросхем).

Конвертор CONV86 позволяет преобразовать исходную программу на языке ассемблера ASM80 в исходную программу на языке ассемблера ASM86.

Машинно-ориентированные фрагменты полученной программы изменяются (переписываются) в зависимости от каждого конкретного случая посредством редактора текстов.

Заключение

Перспективы развития микроЭВМ «Электроника К1»



Выпуск БИС микропроцессорного комплекта К580 поставил задачу разработки средств отладки программного обеспечения и аппаратуры на его основе. Одним из первых в стране было разработано автоматизированное рабочее место программиста на базе микроЭВМ «Электроника К1-10», дальнейшим развитием которой явились микроЭВМ «Электроника К1-30» и контроллер «Электроника К1-20».

По мере развития программных средств и расширения номенклатуры БИС серии К580 появляются мини- и микроЭВМ типа СМ1800, СО-03, СО-04 и т. д., программное обеспечение которых благодаря совместимости операционных систем с микроЭВМ «Электроника К1» может быть использовано на любой из них.

Простота технических решений, широкая номенклатура БИС серии К580, низкая стоимость позволяют использовать их для построения любительских микроЭВМ типа «Микроша», «Радио 85» и др., что привлекает непрофессиональных программистов для дальнейшего развития программного обеспечения и аппаратных средств.

Появление нового 16-разрядного микропроцессорного комплекта серии К1810, программно и аппаратно совместимого с микропроцессорным комплектом серии К580, а также БИС однокристалльных микроЭВМ позволило разработать новое поколение микроЭВМ и персональных ЭВМ.

В настоящее время разработаны миниЭВМ СМ 1810 и четыре типа персонально-профессиональных ЭВМ (ППЭВМ), основой которых стали микропроцессорные комплекты серии К580, К1810, К1816.

Персонально-профессиональные ЭВМ — новый класс

вычислительных устройств, предназначенных для создания базовых средств автоматизации индивидуального труда в широком диапазоне профессиональной ориентации. Это ЭВМ настольного использования, способные работать автономно, в локальных и глобальных сетях, обеспечивают решение широкого круга научно-технических, экономических, административных и других задач. Дальнейшим развитием устройств вычислительной техники явились новые СБИС серий К580, К1810, К1816 — КР580ВВ79, КМ1810ВМ88, КМ1810ВМ89, КР1816ВЕ51, КР1816ВЕ31:

КР580ВВ79 — программируемый контроллер клавиатуры и индикации, позволяющий значительно упростить пульта управления вычислительных устройств, встраиваемых в измерительные приборы и оборудование.

КМ1810ВМ88 — 16-разрядный микропроцессор с байтовым обменом, программно полностью совместимый с микропроцессором КМ1810ВМ86, однако благодаря байтовому каналу данных легко сопрягается с 8-разрядными периферийными устройствами, а также упрощает разводку печатных плат микропроцессорной системы. Применение БИС КМ1810ВМ88 при проектировании одноплатных контроллеров и персональных ЭВМ для индивидуального применения позволит существенно снизить аппаратные затраты.

КМ1810ВМ89 — высокоскоростной двухканальный процессор для организации передачи и обработки информации ввода — вывода, выполняющий с большой скоростью пересылки с прямым доступом, включая пересылки из устройства ввода — вывода в память, из памяти в устройство ввода — вывода, из памяти в память, из устройства ввода — вывода в устройство ввода — вывода. В процессе пересылок процессор может обрабатывать передаваемую информацию: перекодировки, поиск, объединение 2 байт в 16-разрядное слово, разъединение слова на два байта. ПВВ благодаря своей универсальности может выполнять функции: сопряжения шин; обработки строк; управления дисковыми устройствами НМД и НГМД; управления терминалами; передачи информации в многопроцессорных системах.

КР1816ВЕ51 и **КР1816ВЕ31** — высокопроизводительные однокристалльные микроЭВМ соответственно с встроенной и внешней памятью команд. Эти ЭВМ имеют высокую быстродействие; развитую систему команд, включая аппаратное умножение, деление, булевый процессор;

встроенную память команд и данных; два многорежимных таймера-счетчика; пять векторов прерываний; четыре 8-разрядных программируемых канала ввода — вывода; последовательный канал ввода — вывода.

Архитектура и большой набор команд, определяющих в совокупности функциональные возможности однокристальных ЭВМ, позволяют применять их при построении устройств обработки данных различных назначений — универсальных контроллеров, контроллеров периферийных устройств, контроллеров бытовой техники высокой сложности (видеомагнитофонов, телевизоров и т. д.).

Постоянное развитие программных средств микроЭВМ семейства «Электроника К1», как, например, разработка кросс-систем КС48/МС и КС51/МС для однокристальных микроЭВМ КМ1816ВЕ48 и КМ1816ВЕ51, кросс-системы КС89/МС для микропроцессора ввода — вывода КМ1810ВМ89, создание дополнительных модулей, ориентированных на отладку новых типов СБИС в составе аппаратуры, обеспечивают эффективность использования этих микроЭВМ в будущем в качестве отладочных средств программного обеспечения и аппаратуры для перспективных СБИС, совместимых программно с МПК К580, К1810 и ОЭВМ серии К1816.

Приложение

Проектирование интерфейса внешних устройств семейства ЭВМ «Электроника К 1»

Интерфейс внешних устройств предназначен для сопряжения микроЭВМ с внешними устройствами. Примерами внешних устройств микроЭВМ являются: датчики информации; регистрирующие и управляющие устройства; устройства внешней памяти; стандартные периферийные устройства связи с оператором, такие, как перфораторы, фотосчитыватели, дисплеи и т. д.

Исходными данными для проектирования интерфейса внешних устройств являются обычно номенклатура и типы внешних устройств, пропускная способность канала связи.

При проектировании интерфейса внешних устройств решаются задачи хранения информации и обеспечения доступа к ней со стороны процессора; управления обменом; преобразования форматов данных и управления внешними устройствами.

Интерфейс внешних устройств для процессора, выполненного на основе КР580ВМ80А. Структура интерфейса внешних устройств микроЭВМ определяется набором средств, реализующих указанные задачи.

Сопряжение устройства ввода — вывода с системной магистралью микроЭВМ осуществляется с помощью портов ввода — вывода, представляющих собой запоминающие регистры или ячейки памяти со схемами выборки и управления. Каждое устройство ввода — вывода подключается к одному или нескольким портам. Выборка устройства ввода — вывода осуществляется путем выборки соответствующих портов.

Если число устройств ввода — вывода невелико, то для их адресации используется линейный выбор портов, так что в адресном формате отдельные порты ввода — вывода кодируются унитарным кодом, а входы выборки подключаются непосредственно к адресным линиям. При двоичном кодировании адресов внешних устройств используется прямой код. Управление чтением — записью портов осуществляется соответствующими сигналами, поступающими с управляющей магистральной.

В интегральном исполнении порты ввода — вывода реализуются в составе универсальных программируемых БИС интерфейсов, структура и функции которых изменяются под действием программного управления. Программирование осуществляется пересылкой из процессора управляющих слов, для хранения которых в составе БИС вводятся программно доступные регистры управляющих слов. Обращение к регистру осуществляется так же, как и к портам ввода — вывода.

Обмен с внешними устройствами (ВУ) осуществляется тремя способами: программно; по прерыванию; в режиме прямого доступа к памяти.

Программный обмен осуществляется по инициативе процессора и под его управлением. Процессор программно опрашивает состояние ВУ: если ВУ готово, то он выполняет операции обмена — ввода или вывода данных; в противном случае — повторно анализирует готовность ВУ, т. е. ожидает готовности ВУ, выполняя соответствующие команды программ.

Для реализации программного обмена БИС интерфейса содержит порты вводы или вывода данных и порты управления. Порт управления содержит регистр, предназначенный для хранения состояния ВУ, логические схемы выборки и управления.

При вводе данные записываются в порт данных по стробу сопровождения (**ДАННЫЕ ГОТОВЫ**), устанавливается состояние готовности ВУ в порту управления. Процессор читает состояние порта управления с магистралей данных, выдавая его адрес на адресную магистраль по сигналу управления чтением **ЧТВВ**. Аналогично читаются данные с порта ввода данных. По окончании чтения данных состояние готовности ВУ порта управления сбрасывается сигналом **КОНЕЦ ЧТЕНИЯ**.

При выводе внешнее устройство подтверждает прием данных стробом подтверждения **ДАННЫЕ ПРИНЯТЫ**, по которому устанавливается состояние готовности ВУ в порту управления. Процессор записывает данные в порт вывода сигналом **ЗПВВ**. По окончании записи состояние готовности ВУ в порту управления сбрасывается сигналом **КОНЕЦ ЗАПИСИ**. Программный обмен используется для внешних устройств, производительность которых уступает производительности процессора. Недостаток такого обмена — непроизводительная потеря времени на опрос состояния внешнего устройства, что снижает производительность микропроцессорной системы в целом. Пропускная способность канала обмена определяется производительностью внешнего устройства.

Обмен по прерыванию осуществляется по инициативе внешнего устройства. Сигнал его готовности используется для формирования запроса прерывания, который обрабатывается системой прерываний ЦП. Если прерывания разрешены, то осуществляются пре-

рывания программы, выполняемой ЦП, и переход к подпрограмме ввода — вывода.

Интерфейс ввода — вывода по прерываниям содержит порты данных и средства формирования запросов прерываний, которые осуществляют прием и хранение сигналов готовности обмена. Выходы триггеров запросов прерываний непосредственно подключаются к соответствующим входам средств обработки, при этом не требуются цепи сопряжения на выходах (в отличие от используемых при подключении порта управления к системной магистрали при программном обмене). Интерфейс в этом случае упрощается, а функции обмена возлагаются на систему прерываний микроЭВМ.

Обмен по прерыванию позволяет исключить потери времени процессором, затрачиваемые на программный опрос состояния внешнего устройства, и таким образом повысить производительность микропроцессорной системы. Пропускная способность канала обмена в целом такая же, как и при программном обмене.

Обмен в режиме прямого доступа к памяти (ПДП) осуществляется без вмешательства процессора, поэтому для управления магистралью необходимы специализированные средства — контроллеры ПДП, которые совместно со средствами интерфейса ВУ образуют каналы ПДП. Контроллер принимает запросы ПДП от внешнего устройства, формирует сигнал запроса магистрали ЦП и, получив подтверждение об освобождении магистрали от ЦП, берет на себя управление системной магистралью. Он формирует адреса и управляющие сигналы чтения — записи памяти и интерфейса ВУ, а также сигналы отключения интерфейса ЦП от системной магистрали. Цепи интерфейса ВУ формируют сигналы запросов ПДП подобно тому, как формируются сигналы запросов прерывания.

Информация об области памяти, используемой при обмене, в форме начального адреса и размера массива пересылаемых данных загружается в контроллер на этапе его программирования ЦП.

Формирование адресов памяти в режиме ПДП осуществляется адресным счетчиком, содержимое которого увеличивается на 1 после каждого цикла ПДП. Число циклов ПДП контролируется счетчиком, содержимое которого уменьшается на 1 после каждого цикла. По окончании всех циклов ПДП формируется сигнал на выходе КОНЕЦ ПДП, используемый в качестве запроса прерывания ЦП, по которому процессор осуществляет загрузку новых данных в контроллер.

Так как пересылка данных осуществляется непосредственно между ВУ и памятью без затрат времени на выборку и дешифрацию команд, пропускная способность канала обмена в режиме ПДП наиболее высокая по сравнению с пропускной способностью при других способах обмена. Поэтому режим ПДП необходимо использовать для обмена с ВУ высокой производительности. Аппаратурные затраты на

реализацию канала ПДП выше, чем аналогичные затраты при других способах обмена.

Для последовательного обмена требуются средства преобразования последовательных форматов в параллельные.

Управление внешним устройством по программе осуществляется пересылкой управляющих слов через интерфейс. Для реализации программного управления используются порты вывода управляющей информации. Если в интерфейс вводятся аппаратурные средства управления внешним устройством, то он специализируется на управлении этим внешним устройством и называется контроллером ВУ. При подключении внешних устройств, как обычно, необходимо обеспечить требуемую нагрузочную способность выходных цепей интерфейса. Для этой цели могут потребоваться выходные формирователи интерфейса.

Технические средства построения интерфейса внешних устройств для центрального процессора на основе КР580ИК80А. Для построения интерфейса внешних устройств микроЭВМ на основе КР580 используются регистр К589ИР12 и программируемые интерфейсы КР580ВВ55, КР580ВВ51. Контроллер ПДП реализует БИС КР580ВТ57. Регистр К589ИР12 используется для построения простых интерфейсов с большой нагрузочной способностью; БИС программируемых интерфейсов — для организации сопряжения со стандартными протоколами обмена.

Для сопряжения со специализированными внешними устройствами применяются специализированные БИС интерфейса или БИС контроллеров, например: КР580ВВ79 — для сопряжения с клавиатурой и индикацией, КР580ВГ75 — для построения дисплеев на ЭЛТ и т. д.

Интерфейс внешних устройств с параллельными форматами данных реализуется на БИС КР580ВВ55, которая представляет собой однокристальное программируемое устройство ввода — вывода (ПУВВ) параллельной информации различных форматов (программируемый периферийный интерфейс). Программируемый ввод — вывод информации осуществляется по трем 8-разрядным каналам (портам): А, В, С.

В режиме 0 каждый канал может быть запрограммирован на простой ввод или вывод. В режиме 1 восемь выводов каналов А и В могут быть запрограммированы на стробируемые ввод или вывод, при этом выводы канала С используются для приема или выдачи сигналов управления программным обменом по прерыванию. В режиме 2 канал А микросхемы представляет собой двунаправленный 8-разрядный канал данных. При этом пять линий канала С используются для управления (аналогично режиму 1).

Режим обмена задается программно пересылкой на ПУВВ управляющего слова соответствующего формата. Состояние ПУВВ контроли-

руется программно путем считывания слова состояния интерфейса.

Интерфейс внешних устройств с последовательными форматами данных реализуется на БИС КР580ВВ51, которая представляет собой универсальный синхронно-асинхронный приемопередатчик последовательной связи (УСАПП). С ее помощью преобразуется параллельный формат слов данных в последовательный формат со служебными символами для передачи по каналам связи с различными скоростями и, наоборот, последовательный формат принимаемых из канала связи слов данных преобразуется в параллельный формат для ввода в микропроцессор.

Микросхема программируется для работы в пяти режимах: асинхронная передача; асинхронный прием; синхронная передача; синхронный прием с внутренней синхронизацией; синхронный прием с внешней синхронизацией.

Формат передаваемых или принимаемых слов содержит 5, 6, 7 и 8 разрядов данных. Число стоп-битов в асинхронных режимах может быть равным 1, $1\frac{1}{2}$ или 2. В синхронных режимах используется один или два синхросимвола. Микросхема позволяет контролировать ошибки в принимаемых словах данных на четность или нечетность, ошибки в структуре кода при асинхронном приеме, возникающие из-за отсутствия стоп-битов, а также ошибки наложения, возникающие в результате нарушения синхронизации обмена, когда очередное слово поступает до считывания предыдущего слова.

Интерфейс ПДП реализуется с использованием программируемого контроллера КР580ВТ57, предназначенного для организации непосредственной связи между внешними устройствами и памятью. Основная функция — формирование адресов памяти и управляющих сигналов чтения — записи памяти и внешнего устройства. Контроллер принимает запросы ПДП, осуществляет их приоритетную обработку, формирует сигнал захвата магистралей микропроцессору. В результате системная магистраль отключается от ЦП и контроллер формирует последовательность адресов памяти и управляющих сигналов чтения — записи.

Контроллер имеет в своем составе четыре канала ПДП, каждый из которых содержит 16-разрядный регистр, позволяющий адресовать память объемом 64 К байт, и 14-разрядный регистр числа циклов обмена, позволяющий пересылать массивы данных объемом 16 К байт.

Контроллер позволяет осуществлять операции чтения данных из памяти и записи данных в память. Имеется возможность контроля канала ПДП путем выполнения операций ПДП без пересылок данных. Режимы работы и выполняемые функции задаются программно от ЦП. Контроллер имеет два режима приоритетной обработки запросов ПДП — с фиксированными и циклическими приоритетами; два режима синхронизации обмена — обычной и удлиненной записи; режим автома-

тического останова по концу счета; режим регенерации и стыковки массивов данных.

Особенности интерфейса ввода — вывода для центрального процессора, выполненного на основе БИС КМ1810ВМ86. Микропроцессор работает с 8- или 16-разрядными устройствами ввода — вывода. Команды ввода — вывода IN и OUT позволяют адресовать до 64 К регистров устройств ввода — вывода. Устройства ввода — вывода, адресуемые как память, позволяют использовать для операций ввода — вывода систему команд микропроцессора.

По командам IN и OUT данные передаются между УВВ и регистром AX при 16-разрядном обмене или регистром AL при 8-разрядном обмене.

После адресов 0—255 регистров устройств ввода — вывода указывается непосредственно в командах IN и OUT; поле адресов до 64 К указывается косвенно, с использованием регистра DX микропроцессора. Устройства ввода — вывода подсоединяют к мультиплексному каналу микропроцессора или к буферизованным каналам системы; 8-разрядные устройства — к разрядам D15—D8 или D7—D0 канала данных. Подключая равное число устройства ввода — вывода к младшим или старшим разрядам канала данных, достигают равномерного распределения нагрузки. Если устройство подключено к старшим разрядам канала, то всем регистрам присваивают нечетные адреса ($AO=1$). Если устройство подключено к младшим разрядам, то регистры адресуют четными адресами ($AO=0$). Таким образом, адрес определяет передачу 8-разрядных данных по старшим (адрес нечетного байта) или по младшим (адрес четного байта) разрядам 16-разрядного канала данных. Сигнал AO нельзя использовать при адресации регистров внутри устройств ввода — вывода, так как в противном случае не исключаются одновременные передачи по смежным четным и нечетным адресам регистров.

В минимальном режиме микропроцессор формирует сигналы чтения и записи, общие для памяти и устройств ввода — вывода, поэтому если адреса памяти и устройств ввода — вывода перекрываются, то уровень сигнала M/\overline{IO} определяет сигналы выбора устройств. Если поля адресов не перекрываются, то ошибок в декодировании не происходит. При адресации устройств ввода — вывода как памяти декодирование сигналов \overline{RD} , \overline{WR} и M/\overline{IO} обеспечивает отдельные сигналы чтения, записи для памяти и устройств ввода — вывода.

В системах с использованием БИС КМ1810ВМ86 в максимальном режиме контроллер КР1810ВГ88 формирует отдельные сигналы чтения, записи для памяти и устройств ввода — вывода. Управляющие входы памяти и устройств ввода — вывода подключают к соответствующим выходам контроллера.

В системах, выполненных на базе микропроцессора КМ1810ВМ86, возможно применение всех микросхем ввода — вывода серий К580

и K1810. В системах с микропроцессором в минимальном режиме микросхемы ввода — вывода подключаются непосредственно к локальному каналу или к буферу данных.

Тактовая частота на входе устройств ввода — вывода не должна превышать допустимых значений. Временные параметры микросхем ввода — вывода серий K580 и K1810 требуют введения одного такта ожидания в циклах обмена микропроцессора при частоте тактовых сигналов 5 МГц. При меньшей частоте сигналов CLK требуется дополнительный анализ временных параметров для определения необходимости введения такта ожидания в циклах обмена.

Список литературы

1. Крылов Е. И. Однокристалльные микроЭВМ серий К1814, К1820, К1816. — Микропроцессорные средства и системы, 1985, № 2, с. 3—7.
 2. Микропроцессоры. Архитектура и проектирование микроЭВМ. Организация вычислительных процессов/П. В. Нестеров, В. Ф. Шаньгин, В. Л. Горбунов и др.; Под ред. Л. Н. Преснухина. — М.: Высшая школа, 1986.
 3. Коффрион Дж. Технические средства микропроцессорных систем: Практический курс: Пер. с англ. — М.: Мир, 1983.
 4. Микропроцессоры. Средства отладки, лабораторный практикум и задачник./Н. В. Воробьев, В. Л. Горбунов, А. В. Горячев и др.; Под ред. Л. Н. Преснухина. — М.: Высшая школа, 1986.
 5. Средства автоматизации программирования для микропроцессорных комплектов БИС серий К580 и КР580/Н. В. Добровольский, А. В. Кобылинский, В. И. Цыба и др. — Механизация и автоматизация управления, 1983, № 3, с. 22—26.
 6. Кобылинский А. В., Москалевский А. И., Темченко В. А. Однокристалльный высокопроизводительный 16-разрядный микропроцессор КМ1810ВМ86. — Микропроцессорные средства и системы, 1986, № 1, с. 28—33.
 7. Кобылинский А. В., Липовецкий Г. П. Однокристалльные микроЭВМ серии К1816. — Микропроцессорные средства и системы, 1986, № 1, с. 10—19.
 8. Кобылинский А. В., Сабадаш Н. Г., Тесленко А. К. Система автоматизации программирования однокристалльной микроЭВМ. — Микропроцессорные средства и системы. 1986, № 3, с. 23—25.
-

Оглавление

Введение	5
-----------------	---

Глава 1

Элементная база микроЭВМ «Электроника К1»	1.1. Общая характеристика элементной базы рассматриваемого семейства	7
	1.2. Микропроцессорный комплект К580	13
	1.3. Микропроцессорный комплект К1810	30
	1.4. Система команд, программирование микропроцессора	46

Глава 2

Проектирование центральных процессоров микроЭВМ «Электроника К1»	2.1. Проектирование центрального процессора на БИС КР580ВМ80А	77
	2.2. Проектирование центрального процессора на БИС семейства К1810	96

Глава 3

Представители микроЭВМ «Электроника К1»	3.1. МикроЭВМ «Электроника К1-30». Назначение, состав и структура	109
	3.2. Устройство и работа составных частей микроЭВМ «Электроника К1-30»	112
	3.3. Конструктивное исполнение микроЭВМ «Электроника К1-30»	125
	3.4. Контроллер программируемый универсальный «Электроника К1-20»	127

Глава 4

Однокристалльные микроЭВМ серии К1816	4.1. Назначение, структура микроЭВМ семейства, состав функциональных узлов	131
--	--	-----

4.2. Основные режимы работы . .	141
4.3. Система команд микроЭВМ .	146

Глава 5

Программное обеспечение комплекса технических средств микроЭВМ «Электроника К1-30»	5.1. Дисковая операционная система	155
	5.2. Редакторы текстов	167
	5.3. Резидентная система программирования	168
	5.4. Кросс-системы программирования	173

Заключение. Перспективы развития микроЭВМ «Электроника К1»	179
	179

Приложение. Проектирование интерфейса внешних устройств семейства ЭВМ «Электроника К1»	182
---	-----

Список литературы .	189
----------------------------	-----

МикроЭВМ

В 8-МИ КНИГАХ

АЛЬФРЕД ВИТОЛЬДОВИЧ КОБЫЛИНСКИЙ
АЛЕКСАНДР ВАСИЛЬЕВИЧ ГОРЯЧЕВ
НИКОЛАЙ ГРИГОРЬЕВИЧ САБАДАШ
ВАЛЕРИЙ ВАСИЛЬЕВИЧ ПРОЦЕНКО

КН. 3

Семейство ЭВМ
«Электроника К1»

Заведующая редакцией Н. И. Хрусталева
Редактор С. М. Оводова
Мл. редактор Е. В. Судьенкова
Художник В. М. Боровков
Художественный редактор В. И. Мешалкин
Технический редактор Г. А. Фетисова
Корректор Г. И. Кострикова

ИБ № 7152

Изд. № СТД-587. Сдано в набор 06.08.87. Подп. в печать 05.02.88. Т-05059.
Формат 84×108¹/₃₂. Бум. кн.-журн. Гарнитура литературная. Печать офсет-
ная. Объем 10,08 усл. печ. л. 20,58 усл. кр.-отт. 9,53 уч.-изд. л. Тираж
150 000 экз. Зак. № 822. Цена 55 коп.

Издательство «Высшая школа», 101430, Москва, ГСП-4, Неглинная ул., д. 29/14.

Ярославский полиграфкомбинат Союзполиграфпрома при Государственном
комитете СССР по делам издательств, полиграфии и книжной торговли.
150014, Ярославль, ул. Свободы, 97.